Mesure de surface 3D pour la caractérisation ostéo-musculaire de petits vertébrés : application à la caractérisation du vieillissement chez la souris

Estelle Duveau

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministérial : 7 août 2006

Présentée par

Estelle DUVEAU

Thèse dirigée par **Edmond BOYER**
et codirigée par **Lionel REVERET**

préparée au sein **du Laboratoire Jean Kuntzmann (LJK),**
UMR CNRS 5224**.**
et de **l'école doctorale EDMSTII**
Mathématiques, Sciences et Technologies de l'Information, Informatique

Caractérisation squelettique de petits vertébrés par mesure de surface 3D

Thèse soutenue publiquement le **03 Décembre 2012**,
devant le jury composé de :

Saida Bouakaz
Professeur, Université Claude Bernard Lyon 1, Présidente
Laurent Lucas
Professeur, Université de Reims Champagne-Ardenne, Rapporteur
Tamy Boubekeur
Maître de conférence, Telecom Paris Tech, Rapporteur
Marie-Paule Cani
Professeur, Grenoble INP, Examinateur
Edmond Boyer
Directeur de recherche, Inria, Directeur de thèse
Lionel Reveret
Chargé de recherche, Inria, Co-Directeur de thèse

RÉSUMÉ

L'ANALYSE du comportement des petits animaux de laboratoire tels que rats et souris est fondamentale en recherche biologique. L'objectif de cette thèse est de faire des mesures anatomiques sur le squelette de souris à partir de vidéos et de démontrer la robustesse de ces mesures par une validation quantitative. Les principales difficultés viennent du sujet d'étude, la souris, qui, vu comme un objet géométrique, peut subir de grandes déformations très rapidement et des conditions expérimentales qui ne permettent pas d'obtenir des flux vidéos de même qualité que pour l'étude de l'humain.

Au vu de ces difficultés, nous nous concentrons tout d'abord dans le Chapitre 2 sur la mise en place d'une méthode de recalage de squelette à l'aide de marqueurs collés sur la peau de l'animal. On montre que les effets de couplage non-rigide entre peau et squelette peuvent être contre-carrés par une pondération de l'influence des différents marqueurs dans la cinématique inverse. Cela nous permet de justifier que, malgré ce couplage non rigide, des informations sur la peau de l'animal sont suffisantes pour recaler de manière précise et robuste les structures squelettiques. Nous développons pour cela une chaîne de traitement de données morphologiques qui nous permet de proposer un modèle générique d'animation du squelette des souris. La méthode de cinématique inverse pondérée est validée grâce à des vidéos radiographiques.

Ayant justifié de l'utilisation de points à la surface de la peau (l'enveloppe) pour recaler le squelette, nous proposons dans le Chapitre 3 un nouveau modèle de déformation de l'enveloppe. Ce modèle, appelé OQR (pour *Oriented Quads Rigging*, gréage de quadrilatères orientés), est une structure géométrique flexible possédant les bonnes propriétés de déformation de l'animation par cage. A l'instar des squelettes d'animation, elle permet d'avoir une paramétrisation haut-niveau de la morphologie et du mouvement. Nous montrons également comment, grâce à cette bonne déformation de l'enveloppe, nous pouvons utiliser les sommets du maillage déformé comme marqueurs pour la méthode de recalage du squelette du Chapitre 2.

Dans les chapitres 2 et 3, nous avons construit un modèle de souris qui permet d'animer en même temps l'enveloppe et le squelette. Ce modèle est paramétré par OQR. Nous proposons donc dans le Chapitre 4 une méthode d'extraction de ces paramètres à partir soit d'une séquence de maillage sans cohérence temporelle soir directement à partir d'images segmentées. Pour contraindre le problème, nous faisons l'apprentissage d'un espace réduit de configurations d'OQR vraisemblables.

ABSTRACT

ANALYSING the behaviour of small laboratory animals such as rats and mice is paramount in clinical research. We aim at recovering reliable anatomical measures of the skeleton of mice from videos and at demonstrating the robustness of these measures with quantitative validation. The most challenging aspects of this thesis reside in the study subject, mice, that is highly deformable, very fast and in the experimental conditions that do not allow for video data equivalent to what can be obtained with human subjects.

In regards to the first challenge, we first focus on a marker-based tracking method with markers glued on the skin of the animal in Chapter 2. We show that the effects of the non-rigid mapping between skin and bones can be pre-empted by a weighting of the influences of the different markers in inverse kinematics. This allows us to demonstrate that, despite the non-rigid mapping, features on the skin of the animal can be used to accurately and robustly track the skeletal structures. We therefore develop a pipeline to process morphological data that leads to a generic animation model for the skeleton of mice. The weighted inverse kinematics method is validated with X-ray videos.

Chapter 2 proves that points on the surface of the animal (on the envelope) can be used to track the skeletal structures. As a result, in Chapter 3, we propose a new deformation model of the envelope. This model, called OQR (Oriented Quads Rigging), is a flexible geometrical structure that has the nice deformation properties of cage-based animation. Like animation skeletons, OQR gives a high-level representation of the morphology and of the motion. We also show how, thanks to a well-deformed envelope, we can use a sub-set of the vertices of the deformed mesh as markers to apply the method of tracking of skeletal structures developed in Chapter 2.

With Chapters 2 and 3, we have built a model of mice that allows us to animate at the same time the envelope and the skeleton. This model is parameterised by OQR. In Chapter 4, we therefore propose a method to recover the OQR parameters from either a sequence of meshes without temporal coherence or directly from segmented images. To regularise the tracking problem, we learn a manifold of plausible OQR configurations.

CONTENTS

CHAPTER

1

INTRODUCTION.

1.1 CONTEXT - ETHOLOGY

MANY natural and genetically-engineered strains of mice are nowadays used in medical research. It is therefore required to determine the characteristic behaviour, the phenotype, of each strain to be able to determine the effect of the procedures or drugs on the animals during clinical tests. The study of behaviours is called ethology. Most often than not, ethology is conducted by human observers that monitor the animals. This is a time-consuming task that is lacking in several ways. First of all, this greatly reduces the number of animals that can be observed. Secondly, this is often a qualitative evaluation of the behaviour, with no precise measure to objectively describe the phenotype.

In recent years, automatic monitoring of animals through video systems has been increasingly studied. Indeed, such systems make the observation of many animals in an unconstrained environment cheap and non-intrusive. However, automatic extraction of behaviours from such systems is for the moment limited to high-level measures such as the spacial displacement of the animals or a classification of actions such as eating, sleeping, etc [BBDR05]. What is still lacking is a reliable automatic measure of complex behaviours to create strain-specific databases of behaviours.

We therefore propose to try to recover reliable measurable parameters of skeletal structures of rodents through video systems. This would give us an unbiased precise way of studying behaviours. However, rodents, and in particular mice, are highly deformable and move in an unconstrained erratic way, which makes the task very challenging for state of the art motion recovery system usually designed for humans.

1.2 OBJECTIVES

1.2.1 Vestibular control of skeletal configurations in 0G

Our study of rodents is done in collaboration with the CESEM (Centre d'Etude de la SensoriMotricité) research team at Université Paris Descartes. One of their main axis of research is sensorimotor control in vertebrates, that is to say how the central nervous system analyses the different input information it receives from all parts of the body to control the posture and enforce gaze stabilization. This also encompasses the study of balance.

This field of research is of interest for different reasons. For instance, falls are one of the most common causes of accidents in the elderly and better understanding balance impairments may be useful to prevent those falls. Beyond punctual events such as falls, there exists balance disorders such as Ménière's disease that can affect people chronically and become a lifelong disability. On top of clinical applications, the study of balance is also paramount to space conquest. Indeed, astronauts need to be prepared to function in a micro-gravity environment as long space journeys can lead them to experience space adaptation syndrome (also known as space sickness) due to an adaptation of the body to weightlessness.

Balance is a difficult function to study as it is multi-modal. Inputs of different natures are integrated to maintain balance : the visual, somatosensory and vestibular systems all contribute to regulate equilibrium (see Figure 1.1 for a schematic representation). The visual system relies on the scene imprinted on the retina. The somatosensory system gives proprioception, the sense of the relative position and movement of neighbouring parts of the body. The vestibular system detects linear and angular accelerations of the head as well as its orientation with respect to gravity. It originates in the inner ear. All these informations are combined to create a 3 dimensional representation of the position of the eyes, head and body. From this, the central nervous system will coordinate the body to enforce posture and gaze stabilization.

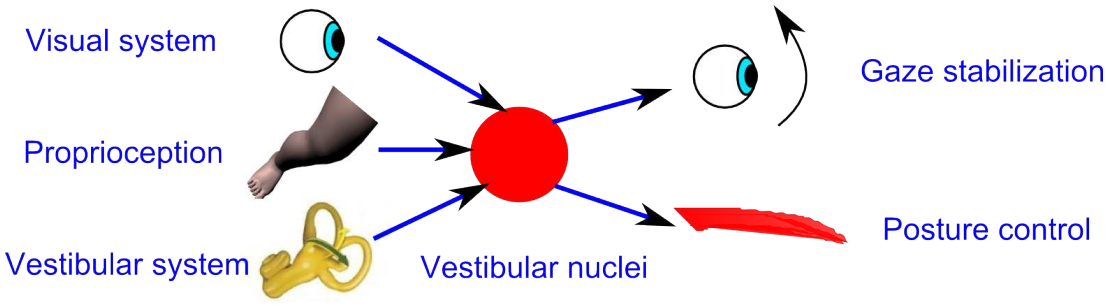


Figure 1.1: *A schematic representation of the inputs and outputs of the central nervous system to achieve balance.*

Our study of rodents aims at better understanding how vestibular control works. This control is complex to analyse as the inner ear is composed of different parts that do not provide the same information. The semi-circular canals (see Figure 1.2) detect rotational movements. It can be noticed on Figure 1.2 that there are three canals that are orthogonal. Each canal detects the rotation in one dimension of space. The otolith organs (see Figure 1.2) sense linear accelerations.

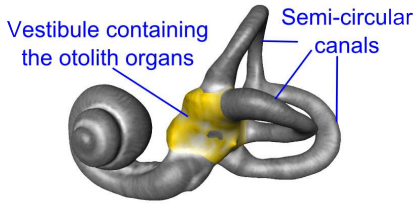


Figure 1.2: *Inner ear - image courtesy of José Braga.*

To study the influence of the vestibular system, mice are submitted to environments of different gravities. Whereas hyper-gravity is easily simulated in a laboratory using centrifuges, micro-gravity can not be simulated in a laboratory. To do so, we took part in parabolic flights in the Zero-G A-300 of CNES (Centre National d'Etudes Spatiales). Parabolic flights give the opportunity to obtain short periods of free fall. On the upward arc of the parabola, the thrust of the airplane is adjusted so that there is no lift. The plane then remains in free fall over the top of the parabola and part of the downward arc. Micro-gravity is thus achieved for about 20 seconds. The flight profile is depicted on Figure 1.3.

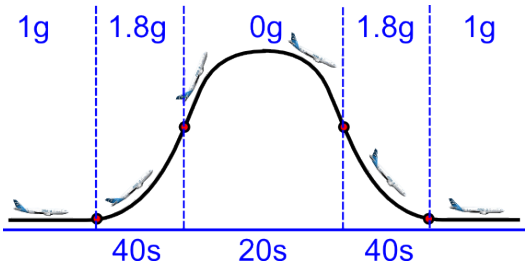


Figure 1.3: *Simplified profile of the parabola described by the plane during a zero-gravity flight. After a period at 1.8 times the gravity (1.8g), micro-gravity (0g) is achieved for about 20 seconds before a new period at 1.8g starts.*

During these zero-gravity flights, three strains of mice were used to compare their stabilizing strategies in a new micro-gravity environment. As well as a control strain

with no vestibular deficiency, ied and IsK mice were used. Ied mice have a deficient otolithic system but functioning semi-circular canals. IsK mice have no functioning vestibular organs at all. As a result, this experiment can help us understand the respective influence of semi-circular canals and otholiths on the skeletal configuration by studying the posture of these different strains of mice experiencing weightlessness for the first time.

1.2.2 Datasets

name	species	motion	number of cameras	number of X-ray videos	presence of markers
ratC0X2M1lo	rat	foreleg during locomotion	0	2	yes - 4 per bone
ratC4X1M1lo	rat	treadmill locomotion	4	1	yes - sparse
ratC6X1M1lo	rat	treadmill locomotion	6	1	yes - sparse
ratC5X0M1sw	rat	swimming	5	0	yes - very sparse
moC8X0M0of	mouse	open field	8	0	no
moC8X0M0pf	mouse	parabolic flights	8	0	no

Table 1.1: *Different datasets used for validation.*

Using mice in zero-gravity flights to develop new techniques of 3D anatomical measurement offers several drawbacks. First of all, experiments in zero-gravity do not allow for a very controlled environment. Indeed, the mice are put in a free-floating space where they can reach any position and orientation. As a result, the cameras can not be located to better capture one given position and orientation but must allow for good capture of all positions and orientations. The mouse therefore occupies only one small portion of the image. The resolution is therefore not optimal. Furthermore, mice are small animals compared to other rodents, for example, rats. Indeed, studying rats instead of mice is easier thanks to the size of the animal. Most importantly, equipments required to validate our techniques can not be incorporated in zero-gravity flights. Indeed, validating measurements of internal structures requires imaging devices such as X-ray videos, which can not be used on-board planes. For all these reasons, we have acquired other datasets that vary in their ability to offer validation and on the controllability of their environment as well as the studied animal. Table 1.1 shows those datasets and their specificities, sorted from the one on which validation of our methods is the easiest to the one where the quality of our measurements can not be quantified

easily. Coincidentally, this also corresponds to the degree of difficulty those datasets offer.

Biplanar X-ray set-up

The bi-planar fluoroscopy set-up at the University of Brown was used to acquire ratC0X2M1lo, 5 bi-planar X-ray sequences (736 frames) of forward/backward motion of the foreleg of a cadaveric rat mimicking locomotion. Figure 1.4 shows the simulated motion. There were 4 markers on the scapula, 4 markers on the humerus, 3 markers on the radius, 1 marker on the ulna and 1 marker on the metacarpal bone of the fourth digit (see Figure 1.5). Thanks to the two X-ray videos, 3D measures can be done on the internal structures and used to validate our results.

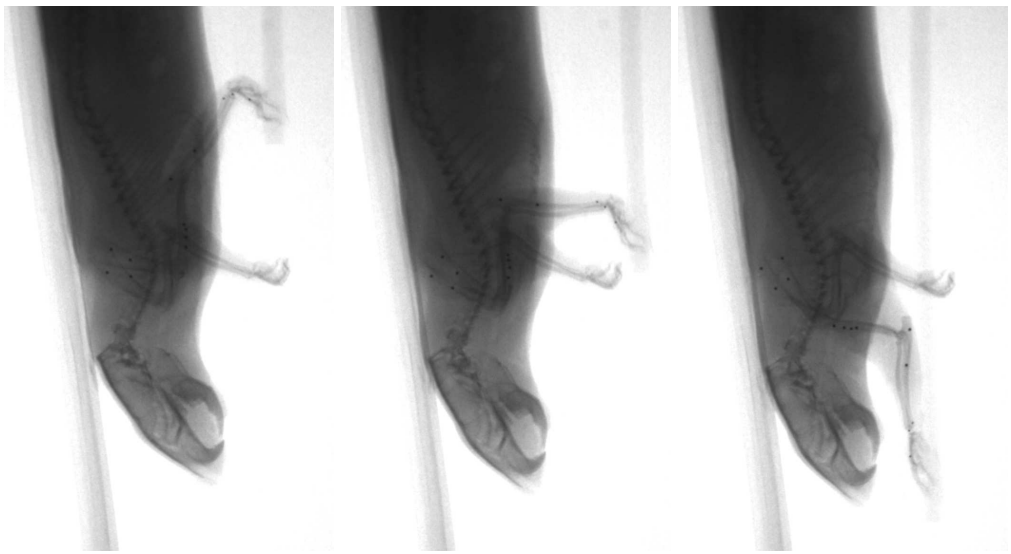


Figure 1.4: *Forward/backward motion of the foreleg of a cadaveric rat.*

Locomotion set-up

In order to validate the measurements of internal structures from external videos for whole morphologies and not just one particular limb, the locomotion acquisition set-up of the Museum National d'Histoire Naturelle of Paris contains one video-radiography to acquire X-ray images of the motion. The acquisition platform is a room with the material required to study the movement of interest. For instance, a treadmill is used to study the locomotion of rodents, a rod is used when studying landing and take-offs of birds. Those equipments also enable us to control the movements of the animals. For the experiments, the set-up contains 4 to 6 standard cameras capturing at 200Hz grey-level images of resolution 640x480. It has been observed that capturing at less than 200Hz leads to losing some information on the movement of rodents as their locomotion cycles are fast. Indeed, in our experiments, a locomotion cycle lasts 250ms

Figure 1.5: *Bi-planar X-ray videos for ratC0X2M1lo.*

in average, captured in 50 frames at 200Hz. Both the video-radiography and the cameras are temporally synchronised through hardware. A validation protocol has been implemented using the visual trace of an analog oscilloscope. Figure 1.6 shows an example of video acquisition for rat locomotion. One can notice that the experimental conditions are hard to set. In particular, getting the animals to stay within the captured space of all cameras without constraining its natural movement is a challenge.

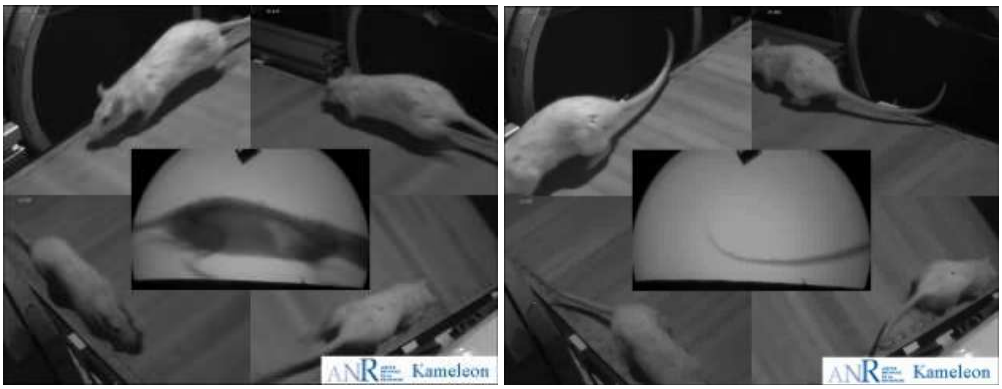


Figure 1.6: *Set-up for the study of locomotion of rodents. The animal walks on a treadmill in front of an image intensifier (X-rays) and 4 to 6 standard cameras. The lack of constraints of the environment leads to periods of time where the animal is not completely covered (right).*

Once the platform is set up, the goal is to be able to find the 3D coordinates of a point from its projections on the images. To do so, the cameras must be calibrated in the same reference frame i.e. their intrinsic parameters must be computed as well as

their position and orientation in a common reference frame. This includes the video-radiography. When using standard cameras, distortion is considered negligible. However, for video-radiography, the distortion produced by the cameras is modeled as a radial and a tangential distortion [Bro71]. This distortion is thus represented by the distortion coefficients used to map a point from its observed position to its corrected position. All X-ray images are thereafter treated to correct distortion before being processed. Figure 1.7 shows an example of correcting distortion.

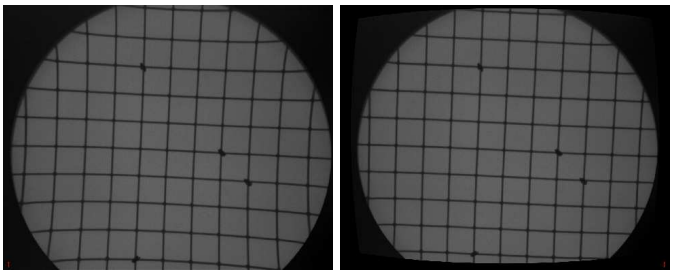


Figure 1.7: *Image with distorsion (left) and same image after calculation of the distorsion coefficients and distortion correction.*

Once the distortion is corrected, the intrinsic and extrinsic parameters of the cameras and X-ray video are computed using [Zha00]'s implementation in OpenCV [Bra00]. As the video-radiography has to be calibrated in the same reference frame as the cameras, the calibration objects, more precisely their features used to calibrate the cameras, must be visible both by camera and X-ray imaging. On top of that, the use of X-rays prohibits the presence of humans within the set-up to manipulate the calibration objects. Such a calibration object of known dimensions can be seen on Figure 1.8 as well as the support used to be able to set it in different positions without the need for someone to hold it. Figure 18 also shows the common reference frame obtained, valid for both cameras and X-ray videos.

As a result, this set-up, used with 6 standard cameras for sequence ratC6X1M1lo and 4 standard cameras for sequence ratC4X1M1lo to study rat locomotion, can be used to measure the 3D positions of points visible on the standard cameras as well as their projections on the X-ray video.

Swimming set-up

For a more challenging motion, we have also acquired swimming motions of rat. Figure 1.9 shows one frame of ratC5X0M1sw. However, due to water distortion, it is harder to obtain precise measure on the images.

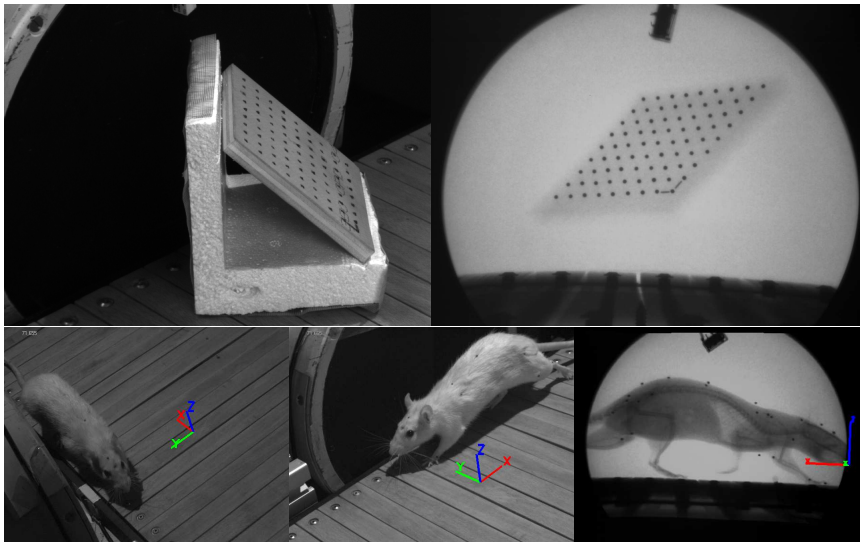


Figure 1.8: *Object of known dimensions used for calibration and result of the calibration of both the cameras and the X-ray camera in a same 3D reference frame represented by the X,Y,Z vectors.*

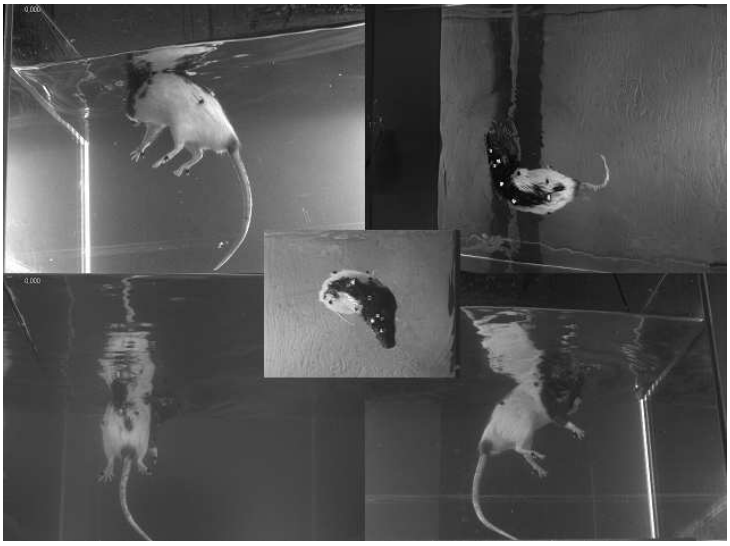


Figure 1.9: *Set-up for the study of swimming of rodents. Notice how the water creates distortion.*

Open field set-up

Unlike the previous experiments, an open field set-up captures a large space on which an animal can do as it pleases. For instance, in sequence moC8X0M0of, mice are free to roam on a flat square, unlike in the locomotion sequences where the animals where constrained by the treadmill width and speed (see Figure 1.10). In these sequences,

markers were not used, which makes quantitative validation possible, but not qualitative validation.

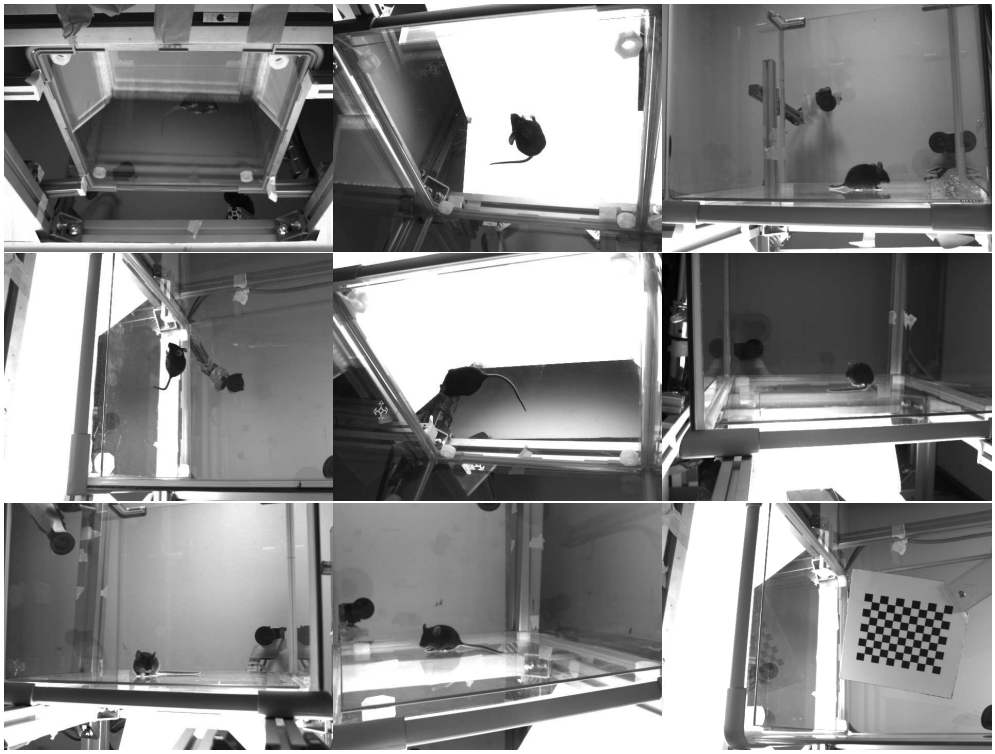


Figure 1.10: *Set-up for open field study of mice : 8 cameras and an example of calibration object.*

Parabolic flight set-up

To analyse the behaviour of the different strains of mice during parabolic flights, we designed a free-floating set-up as a 50cm^3 cube in which the mice can float. The design and building of the set-up was a fastidious and time-consuming task as the set-up had to adhere to a number of mechanical and electrical rules for safety reasons as well as for the fact that there could be no trial. A camera was installed in each corner of this cube, creating a multi-view set-up of 8 high-speed synchronised grey-scale cameras. The cameras were Point Grey Research Grasshopper with a 640x480 pixels resolution, 7.4μm CCD captors running at 120Hz with objectives of 5mm focal length. The cameras and the acquisition system were provided by 4d View Solutions [4dv]. We then designed and built an air-tight rig especially engineered to resist the extreme conditions of a zero-gravity flight in which to encase the monitored free-floating system. Images of the set-up and the free-floating space can be seen on Figure 1.11.

As can be seen on the right of Figure 1.11 (green block), there are 3 cages to hold the

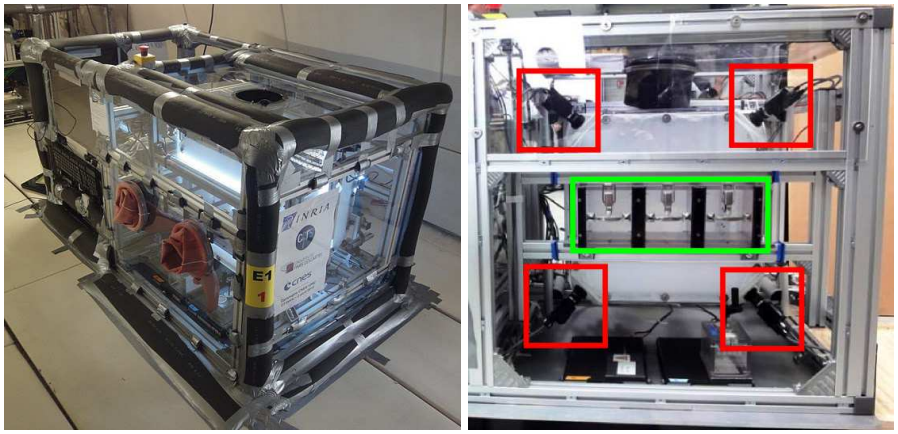


Figure 1.11: *Parabolic flight set-up : (left) the whole set-up - (right) the free-floating space monitored by cameras.*

mice out of the free-floating space. During the experiment, each one holds a mouse of a different strain. During a flight, six series of five parabolas in a row are experienced with about five minutes in between two series. As a result, using the manipulating gloves (seen in pink on the left of Figure 1.11), each mouse can be moved from the holding cage to the free-floating space in-between series so that the different strains can be tested without breaking the requirement of an air-tight set-up.

The different parabolas are recorded by the cameras. The cameras are placed according to two criteria : maximising the acquisition space i.e. the space covered by cameras and maximising the viewpoints i.e. not privileging any view direction as mice in 0g can reach any position and rotation within the free-floating space. As illustrated on Figure 1.12 and Table 1.2, by placing the cameras at each corner of the free-floating cube aiming at the opposite corner, 18.2% of the space is covered by all cameras, 56.2% of the space is covered by at least 4 cameras and 96.5% by at least 2 cameras.

Number of cameras	Space covered exactly	Space covered at least
8	18.2%	18.2%
7	7.1%	25.3%
6	7.5%	32.8%
5	6.8%	39.6%
4	16.6%	56.2%
3	14.0%	70.2%
2	26.3%	96.5%
1	3.5%	100%

Table 1.2: *Percentage of the free-floating space covered by exactly (middle column) and at least (right column) a given number of cameras (left column).*

The cameras are synchronised and calibrated. The synchronisation is done through

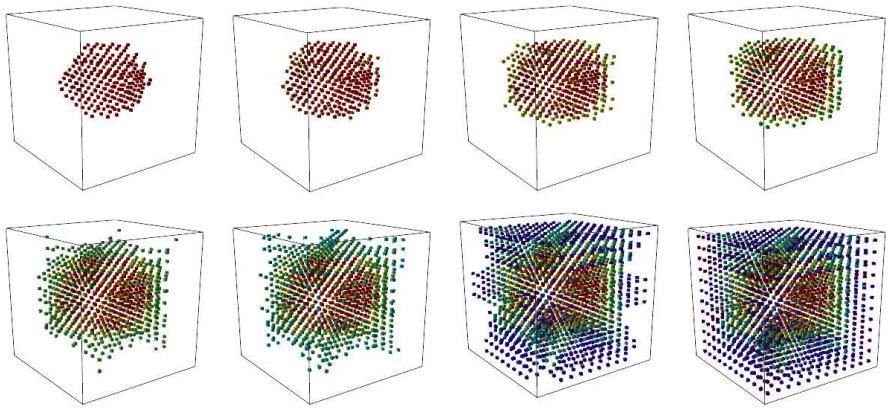


Figure 1.12: *Representation of points within the free-floating space that are covered by at least (from left to right) 8, 7, 6, 5, 4, 3, 2 and 1 camera.*

the 4d view system [4dv] and calibration is done pre and post-flight using manufactured objects of precise dimensions using [Zha00]. Doing the calibration before and after the flight enables us to check if the plane vibrations and changes in gravity have displaced the cameras. If this is the case, the black-and-white targets sticked to the cage that can be seen on the mounts of the free-floating cube on Figure 1.13 are used to compute at each frame the homography between the current image and the images used at calibration. A homography or projective transformation is an invertible linear transformation from a projective space to itself. It preserves colinearity and incidence. Intuitively, it describes what happens when the point of view of the observer changes. Applying the homography to the 2D points used for computing the extrinsic parameters enables us to compute the correct location of the camera at each frame. The calibration objects can be seen in Figure 1.13. Once the intrinsic and extrinsic parameters of the cameras are computed, the imaging parameters and the position and orientation of the different cameras are known in the same reference frame and 3D measures can be made in the free-floating space. All the sequences acquired with this set-up are labeled moC8X0M0pf.

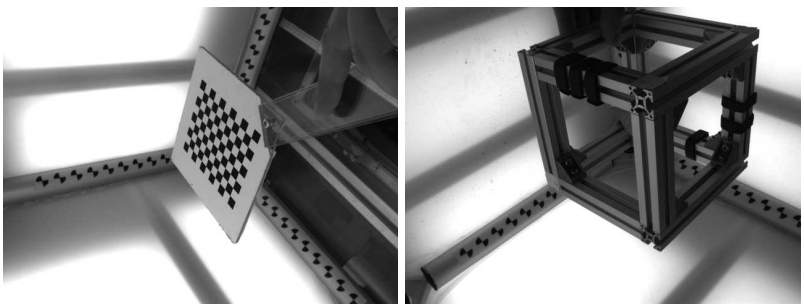


Figure 1.13: *Calibration objects : the checker is used for intrinsic parameters and the cube for extrinsic parameters.*

1.3 Overview and Contributions

1.3.1 Goals

Our objective is to find robust measurements of behaviours of rodents. The requirement for robustness means that the evaluation must be quantitative and thorough. As we saw in the previous section, we have built and used different set-ups that make tracking increasingly difficult. Those different set-ups also allow for various degrees of validation.

Our goal is therefore to develop a model of rodents and a tracking method to register this model to videos that can be used for all input data. Indeed, we do not want to have to build a new model for every experiment. We therefore choose to include all skeletal structures in our model to be all-encompassing. However, depending on the input data, we want to evaluate which measures are robust : are we able to robustly recover the position and orientation of each bone or can we only trust our measure of the center of mass of the animal?

1.3.2 Contributions

In this perspective, we propose a new animation model : **Orientation Quads Rigging** (*OQR*). OQR combines a high-level representation similar to animation skeletons with the smooth, realistic and more flexible deformations of cage-based animation. It can be used for both articulated and non-articulated animation. We use it to model and deform the skin of rodents in a realistic way. The skin is then used to recover the skeletal structures through weighted Inverse Kinematics (IK). A set of skin features proved to lead to robust and accurate IK is used as markers.

We also develop a new motion recovery method based on manifold learning of OQR. Indeed, to constrain the tracking to plausible postures of the rodent, we search for the deformation in a low-dimensional subspace of OQR configurations. Using the OQR representation allows for a more flexible model that adapts more robustly to noisy observations compared to animation skeletons. Figure 1.14 illustrates the process.

1.3.3 Overview

Being able to infer the skeletal structures from the deformation of the skin is the grounds of our studies. Section 2 presents the animation model for the skeletal structures (Section 2.3) and the weighted Inverse Kinematics method used to track the skeleton from features on the skin (Section 2.4). We validate this method by measuring errors in 3D on the bi-planar X-ray sequence ratC0X2M0lo and errors in 2D on the treadmill sequences that have one X-ray video (ratC6X1M1lo, ratC4X1M1lo).

As we can recover the skeletal structures from the skin by tracking features on the

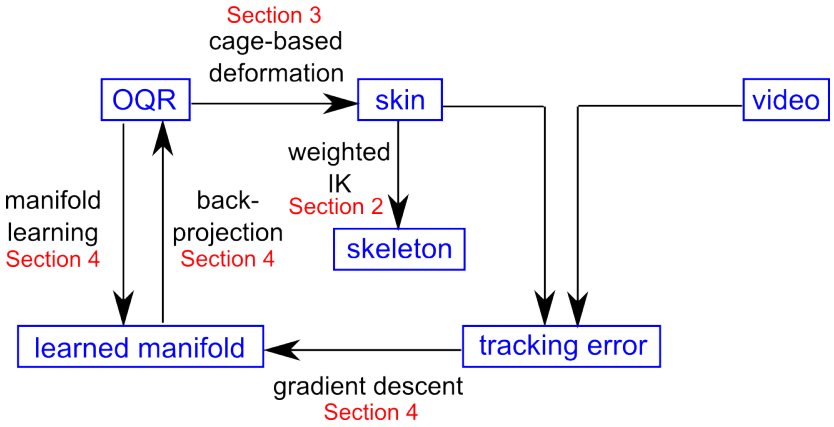


Figure 1.14: *Overview of our contributions.*

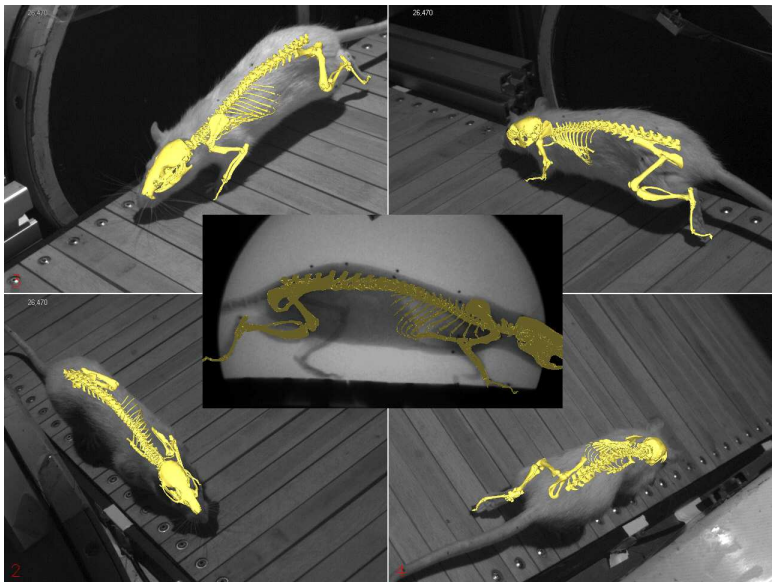
skin, Section 3 develops a deformation model for the envelop of the animal. This new animation model called Oriented Quads Rigging (OQR) allows for smoother, more realistic and more flexible deformations of rodent skin compared to skeleton-based skinning while retaining a high-level parameterisation of motion (Section 3.3). Section 3.4 shows how this model is used in the framework of Section 2 to recover the skeleton from the OQR configuration.

An animation model of rodents including both the skeleton and the skin is therefore built in Section 2 and Section 3. Section 4 presents the tracking method used to recover this model from multi-view inputs. This approach is a flexible model-based approach that allows for the recovery of OQR from a sequence of 3D meshes without temporal coherence (Section 4.3) or directly from segmented videos (Section 4.4). To regularize the tracking process, the OQR configuration is constrained to plausible poses using manifold learning (Section 4.2). Results are shown on sequences ratC6X1M1lo (Section 4.3.3 and Section 4.4.3), moC8X0M0of (Section 4.3.3) and moC8X0M0vp (Section 4.4.4) as well as on standard human datasets for comparison with state of the art motion recovery techniques. Depending on the input data, the possibility of quantitative validation varies as well as the level of details that can be robustly tracked.

CHAPTER

2

SHAPE AND MOTION OF THE SKELETAL STRUCTURES.



THE need to accurately measure internal structures has been studied in many field of studies such as biomechanics, computer graphics or biology. However, most techniques for 3D acquisition of moving skeletons require expensive cumbersome experimental set-ups. The experimental process is also often invasive. While acceptable for animals, it becomes difficult when dealing with animals. This chapter develops a marker-based method to measure the 3D skeletal postures of animals that loosens the experimental constraints in both the type and cost of equipment required and the invasiveness of the process to the subjects of the study, rodents in our case. Our motion analysis method relies on both morphological and motion data acquisition (Section 2.2). The morphological data acquisition results in a set of polygonal meshes representing the anatomical skeleton. The motion data acquisition results in 3D trajectories of markers. An animation model based on a set of articulated solids is then built from the polygonal models of the studied bones (Section 2.3). Then, the animation model and the trajectories of markers are combined to animate the polygonal models of the bones through inverse kinematics (Section 2.4). Our inverse kinematics system enforces two types of constraints : an anatomical one and a tracking one. This method is automatic and robust enough to be applied on different morphologies and movements (Section 2.5).

2.1 PREVIOUS WORK

2.1.1 Internal vs External 3D imaging

Understanding the functions of skeletal structures of vertebrates in motor control requires accurate 3D measures of posture and movement of the skeleton. [CFM+08] shows the shortcoming of 2D analysis in rat gait analysis. Visualization of animations of 3D bone models can also be an helpful way of presenting data. In [ZNA03], it is used to diagnose low back pain in humans. However, current techniques for 3D acquisition of skeletons such as CT-scan operate in a static way only. Relying on computer vision methods, different types of imaging techniques can be used to acquire the necessary dynamic data of moving skeletal structures. They can be classified as internal and external imaging. Internal imaging such as X-rays provides direct images of the skeletal structures (see Figure 2.1). At least two of these views have to be used to be able to extract 3D information from the 2D data as in [BZBT06], [BKZ+07] and [DSG+04]. More than two views increases the accuracy of the 3D measurement but these equipments are usually expensive, of considerable size and not trivial to handle. [WB00] explains all the different parameters that can be tune as well as the danger of extensive exposition to X-rays. All this sometimes prohibit the use of an X-ray image intensifier, let alone the use of two of these devices simultaneously.

Still, the requirement of two internal imaging equipments can be overcome. Adding constraints such as collision detection can lead to enough constraints to locate the bones, e.g. for human knees in [PKS+09]. However, it becomes computationally

prohibitive for whole morphologies. In the particular case of human study, only one C-arm can be used if the subject can hold the studied position while the C-arm is being moved [AWM+11]. However, such an experimental process is not possible with animals. One way to get around that issue is to apply Dynamic Time Warping (DTW : see [Sak78]) to the data. This technique allows the extraction of 3D information from a 2D set-up by temporally aligning data from two different acquisition sequences of the same movement. Typically, this is done with a lateral view and a dorso-ventral view (see Figure 2.1). Once the two sequences are aligned, we can do as if we were working on a set-up with multi-view internal imaging. However, the application is limited to movements that can easily be reproduced within a certain accuracy. It is therefore not applicable outside regular locomotion on a treadmill.

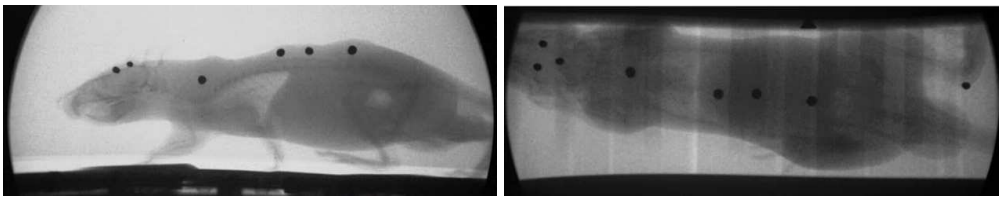


Figure 2.1: *Example of two different acquisition sequences used for Dynamic Time Warping.*

External imaging such as cameras are less invasive and less difficult to set-up. Nowadays, a multi-camera set-up can easily be installed for a small cost. However, external imaging gives an indirect access to the skeletal structures because what is being observed is the enveloppe of the animal. This enveloppe is not rigidly connected to the skeletal structures as muscles and skins are not rigidly connected to bones (see Figure 2.2). [BC10] and [FPC+06] have measured the bias generated by assuming that a marker on the skin is rigidly connected to an internal structure during rat locomotion. As it is non-negligeable, measures relying on external imaging only are not as precise and accurate as measures relying on internal imaging.

2.1.2 Marker-based vs Markerless tracking

From this internal or external videos, different features can be extracted and used for tracking. Some techniques use markers, others rely on other cues. Markerless methods can either be automatic or manual. Automatic image-based registration ([BZBT06], [BKZ+07], [DMKH05]) is a very efficient way to process data but is unfortunately sensitive to segmentation noise and not yet available for whole-body tracking. All the studies mentioned above focus on only one articulation (see Figure 2.3). If automatic registration can not be used, scientific rotoscoping, introduced in [GBJD10], is the most precise method. An articulated model is registered manually to fit the images (Figure 2.4). Not all images have to be specified as the animation curve can be interpolated in-between key-frames. However, this is still a very tedious and time consuming

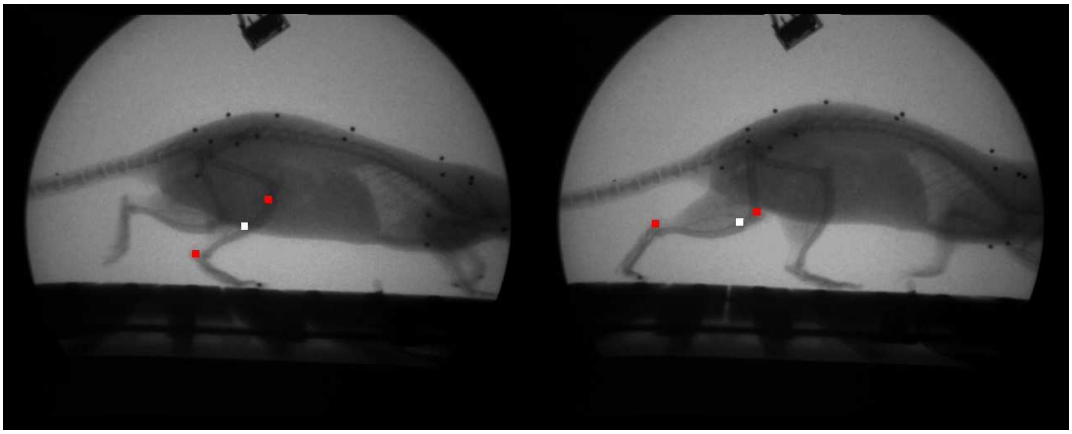


Figure 2.2: *Skin sliding effect : the markers in red are surgically implanted directly on the bones (at the ankle and at the knee) whereas the white marker is glued on the skin of the tibia.*

method as the accuracy is as good as the user wants it to be. As a result, markerless techniques for internal measures require too much user input and clear imaging to automatically study the motion of an entire morphology.

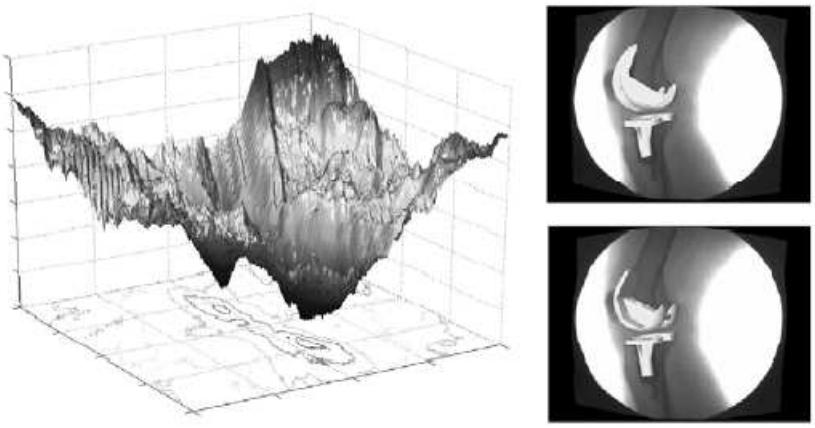


Figure 2.3: *Example from [DMKH05] of automatic registration - left : objective function values - right : corresponding poses for the two largest minima.*

On the other hand, marker-based techniques have been used intensively in biomechanics, biology and computer animation for automatic measure of the entire skeleton. While computer animation focuses on fast automatic methods to recover a natural pose for the body, at the expense of accuracy, biology usually focuses on a precise and accurate measure for a few bones. Depending on the imaging technique used, one can

Figure 2.4: *Example from [GBJD10] of scientific rotoscoping. The different bones are registered manually.*

use either internal or external markers. Markers implanted on the skeletal structures, as in [BBG+10] or [TA03], are rigidly connected to its associated bone but requires a more invasive process. Implanting markers on some of the thin long bones of small animals can indeed be a difficult surgery. External markers are easier and less invasive to implant but as the skin is not rigidly connected to the bones, neither are the markers, especially in widely studied animals such as rats and mice where skin-sliding artefacts are important, as demonstrated in [BC10] and [FPC+06]. As a result, the link between the position of the markers and the bones is not trivial.

2.1.3 Inverse Kinematics

Once the 3D trajectories of the markers are extracted from the videos, different tracking techniques can be used to extract 3D kinematic information about the skeletal structures. A common technique consists in computing the transformation of a set of markers considered rigidly connected to the same bone. The transformation of this set from one time step to another gives the transformation of the underlying bone. An example from [BBG+10] can be seen on Figure 2.5. This method is accurate and precise. However, it requires at least 3 non-colinear markers to be rigidly connected to each other and to each of the skeletal structures of interest. It can be difficult to attach that many markers on small animals, firstly for practical reason, secondly given the limited resolution of imaging devices.

Decreasing the number of markers required is possible as proven with techniques such as inverse kinematics. Indeed, most computer animation techniques consider a simplified version of the skeleton as a set of rigid bodies with a reduced number of degrees of freedom. Usually, the skeleton is a hierarchy of rigid bodies linked by ball-and-socket joints. Some joints are sometimes further constrained to only have one or two degrees of freedom. For instance, the elbow is usually modeled as a hinge joint, having only

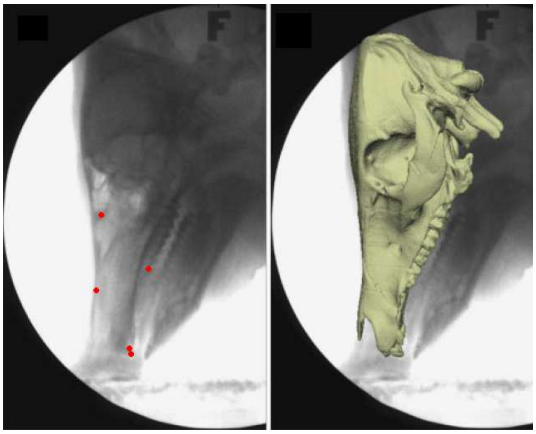


Figure 2.5: *An example from [BBG+10] of rigid body motion recovered by computing the rigid transformation of a set of markers rigidly connected.*

one degree of freedom allowing flexion and extension. As a result, computer animation usually uses a simplified skeletal animation model instead of independent animation of the different bones. This allows for techniques that require less input to produce an animation of the different structures of the skeleton. The most common one is inverse kinematics [ZB94]. Inverse kinematics solves for the degrees of freedom of the system of rigid bodies so that a desired position of an end-effector is reached. However, inverse kinematics does not account for the fact that in experimental conditions, data can be noisy or incomplete.

To be more robust to noise, [ZVDH03] proposed to use the trajectories of the markers as an attractor to landmarks on physical human models. Virtual springs are created between the markers and the landmarks and torques are applied at the joints of the model. A physical simulation then finds the body posture that penalizes gaps between markers and landmarks. However, all these inverse kinematics technique only give a simplified model of the anatomical skeleton. For instance, the spine is usually modeled by a couple of rigid bodies in humans, limiting the range of possible motion. To reach a higher level of details and accuracy, one has to resort to model-specific procedural methods. In [UPBS08], a sequential human-specific inverse kinematics system is used for fast, detailed and precise motion capture of humans from six markers.

2.1.4 Overview of our method

Existing methods to register animal morphologies on videos require invasive and tedious experimental constraints or lack accuracy and automatism. As [UPBS08] did for humans, we propose a complete motion capture pipeline for rodents that loosens the experimental constraints while producing simulations of the same accuracy as existing methods. In the prospect of screening animals in laboratories, we focus our study on the locomotion of the rat. Figure 2.6 illustrates this process. The pipeline has two main

branches, one corresponding to the acquisition of the morphology (Section 2.2.1) and the animation model built from it (Section 2.3), the other to the acquisition of motion data (Section 2.2.2). Those two branches join to produce the animation of the internal structures from the videos (Section 2.4).

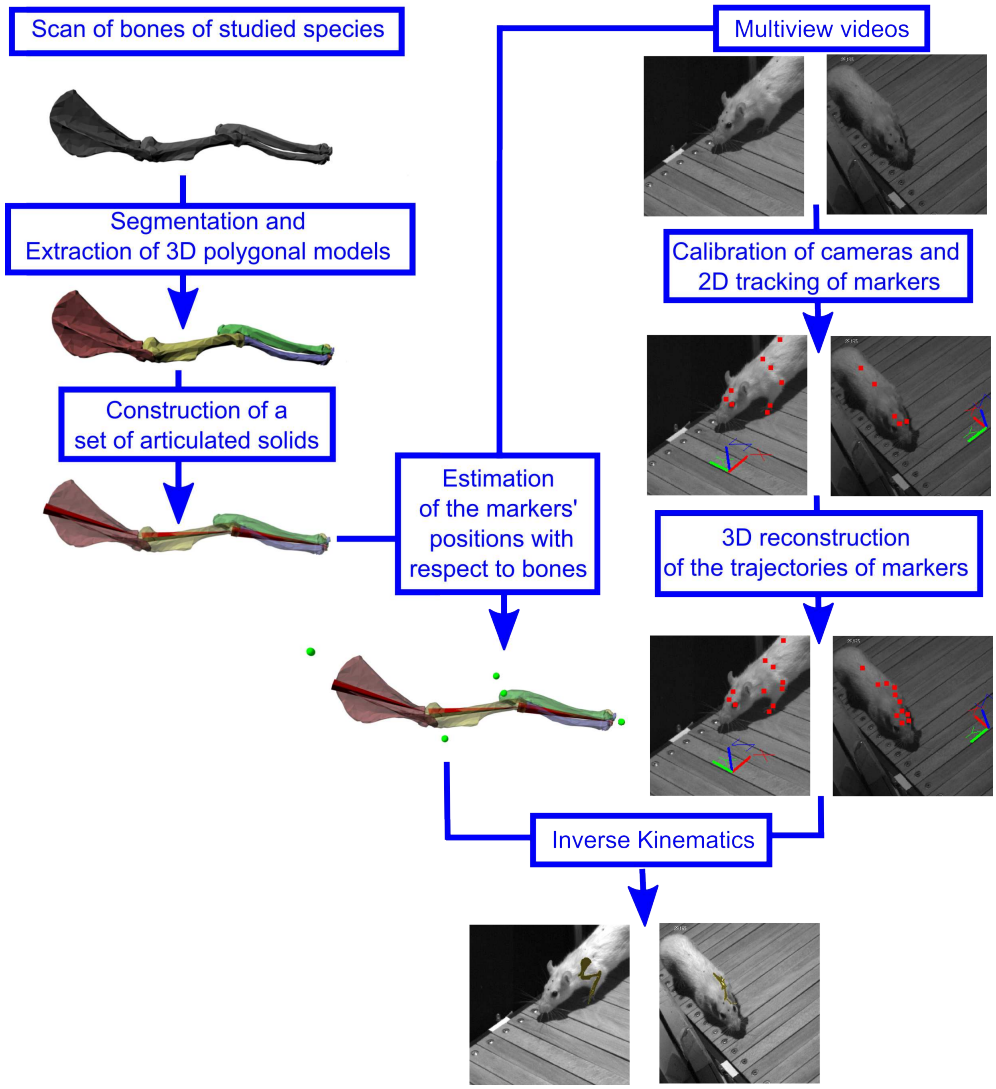


Figure 2.6: *Pipeline of our method for marker-based tracking of moving morphologies.*

2.2 GEOMETRY AND MOTION DATA ACQUISITION

To animate animal morphologies from videos, one must first acquire both the morphology of the animal and some motion cues from the videos, as illustrated by the two branches of Figure 2.6. The animal morphology data is represented by a set of 3D polygonal models extracted from CT-scans. The motions cues from the videos are the 3D trajectories of markers implanted on the skin of the animal.

2.2.1 Acquisition of the geometrical models of the bones

The geometrical models of the bones of the animal to be studied are extracted from Computed Tomography scans (CT-scan). The CT-scan of the animal gives us volumetric data where each element of the volume, each *voxel*, contains a density value. The higher the density of the material being imaged, the less X-rays make it through to the film. As bone is denser than flesh, bones are displayed in white in CT-scans whereas flesh is displayed in grey (see Figure 2.7). If one takes into consideration only the voxels whose density is the same as the density of bone, one can extract the skeleton from the CT-can (see top left of Figure 2.8). From these voxels, an isosurface can be built with a Marching Cube algorithm [LC87]. This results in a 3D mesh (see top right of Figure 2.8).

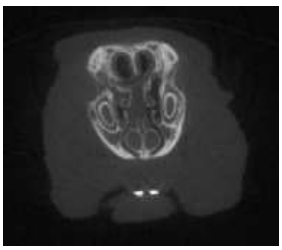


Figure 2.7: *Slice of a CT-scan of the skull of a rat. The bone's density is higher than that of soft tissues, which means the skull is displayed in white whereas the flesh is displayed in grey.*

However, this 3D mesh is one non-articulated piece containing all bones, without any way of differentiating between the different structures. Using available volumetric visualisation softwares such as Amira Software (Visage Imaging), the different skeletal structures can be segmented on the volumetric data. Once the segmentation is done, the 3D polygonal models of the different bones can be extracted as separate meshes though Marching Cubes. Figure 2.8 shows on the bottom row the result of segmentation by colouring each segmented structure differently.

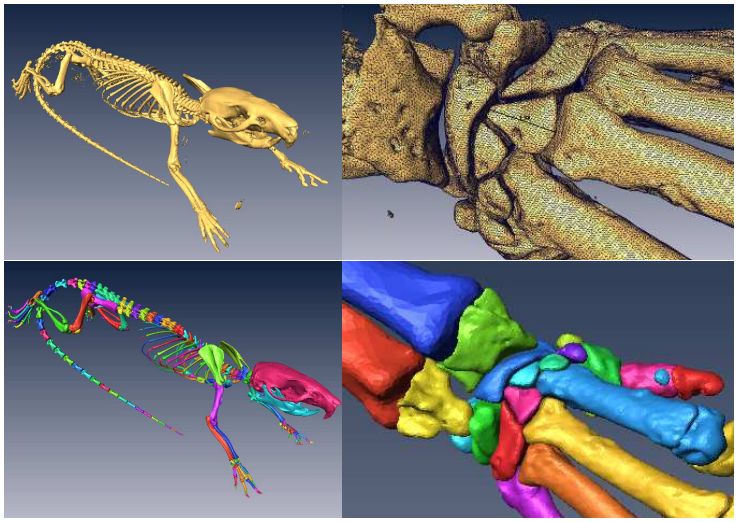


Figure 2.8: *CT-scan of a rat - top : unsegmented isosurface of the volumetric data - bottom : segmentation of the different bones where each bone is represented by its own colour.*

In our case, rodents are rather small animals. To be able to properly segment and extract the different bones, we had the opportunity to acquire high-precision and high-resolution scans at the European Synchrotron Radiation Facility (ESRF) in Grenoble. This acquisition resulted in 4 CT-scans with a 50 micrometers precision of 4 rats. On the right of Figure 2.8, we can see the level of details we can obtain on the front paw of the rat. Such precision allows us to individually segment the carpal bones of the rat. However, some other CT-scans we acquired did not have a good enough precision to be able to segment all the individual bones. Using [GRP10], we were able to adapt the ESRF CT-scan to automatically segment low-resolution CT-scans of both rats and mice. Figure 2.9 shows an example of automatic segmentations of another rat.

We now have acquired the morphological information we need to register morphologies to videos in the shape of a set of 3D meshes, each mesh representing one individual bone. The motion information used to register those meshes to the videos is the 3D trajectories of markers implanted on the animal.

2.2.2 Acquisition of the trajectories of the markers

As explained in Section 2.1, markerless techniques for internal measures require too much user input and clear imaging to automatically study the motion of an entire morphology. We therefore first focus on marker-based methods. As a result, the information extracted from the videos are the trajectories of markers.

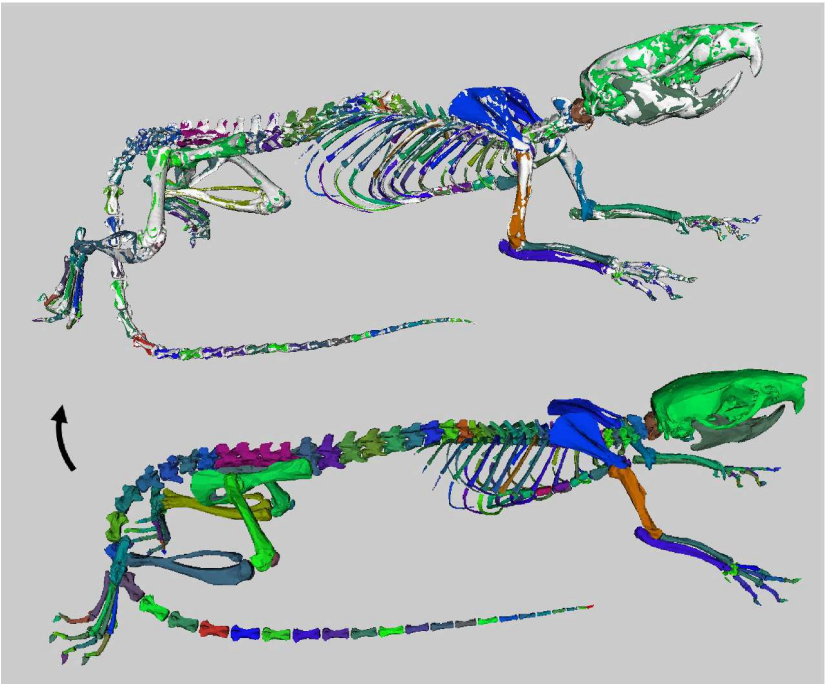


Figure 2.9: *From [GRP10], automatic registration of the segmented model of Figure 2.8 (bottom) to a surface extracted from another CT data (top in white).*

As all the cameras of the set-up are synchronised and calibrated in the same reference frame, the set-up can be used to measure the 3D position of points from at least 2 of their projections by triangulation. Indeed, knowing the imaging parameters as well as the position and orientation of a camera, the pixel coordinates of the projection of a 3D points gives us a line on which the 3D point lies (see Figure 2.10), the viewing line. Two projections of the same 3D point on two different viewpoints give two viewing lines which intersect at the 3D point.

In our case, the 2D projections of the markers are tracked through the sequence automatically (several methods can be used : Lucas-Kanade, correlation factor, see [Hed08] for a short survey). If automatic methods fail on some frames, manual tracking can be made less difficult by using *epipolar geometry*. Epipolar geometry gives a number of geometric relations between the 3D points and their projections onto the 2D images that lead to constraints between the image points. For example, given the projection of a 3D point on a camera, the corresponding viewing line can be projected on another viewpoint (see Figure 2.11). This projection is a line called the epipolar line of the 2D point and its equation can be directly computed from the camera parameters. The projection of the 3D point on this new viewpoint is therefore known to be on the epipolar

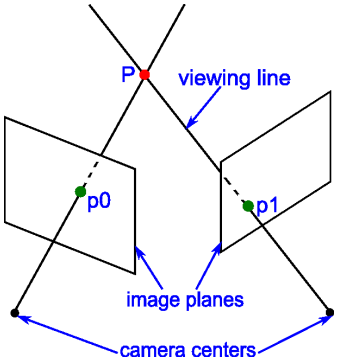


Figure 2.10: *Triangulation : if the projection of a 3D point P is known in two viewpoints : $p0$ and $p1$, then the position of P can be built as the intersection of the 2 viewing lines.*

line. Figure 2.12 shows an example on rat locomotion. Figure 2.13 shows an example of gait analysis of a rat by showing the 3D trajectories of the eyes and left toe of the animal during locomotion.

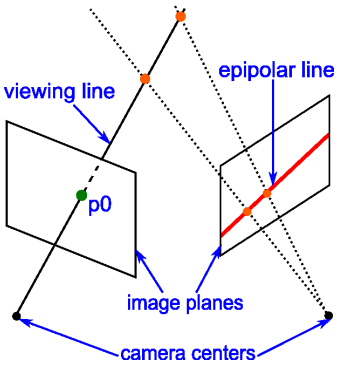


Figure 2.11: *Epipolar geometry : if the projection of a 3D point is known to be $p0$ in a viewpoint, the projection in another viewpoint lies on the epipolar line of $p0$ (red line).*

We have thus obtained morphological information from CT-scans and motion information from videos. The morphological information is a set of 3D meshes and the motion information is a set of 3D trajectories of markers. To be able to register the meshes on the markers, one must first define how the 3D meshes are allowed to move.

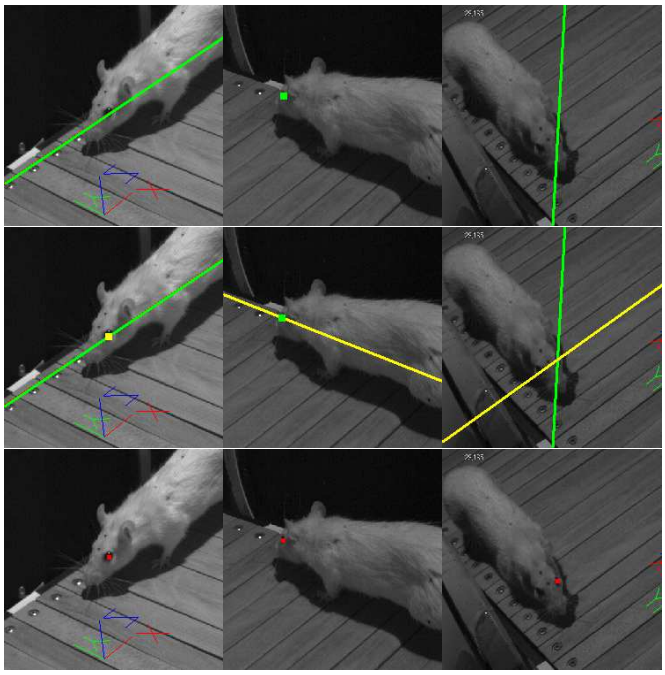


Figure 2.12: *Example of marker tracking - top row : a point is marked on one camera (green point), the epipolar lines can be computed for the other viewpoints (green lines) - middle row : the same point is marked on another camera (yellow point), it lies on the green epipolar line. The epipolar lines can be computed for the other viewpoints (yellow line) - bottom row : from those two projections, the 3D coordinates of the marker can be computed and projected back to the cameras (red points).*

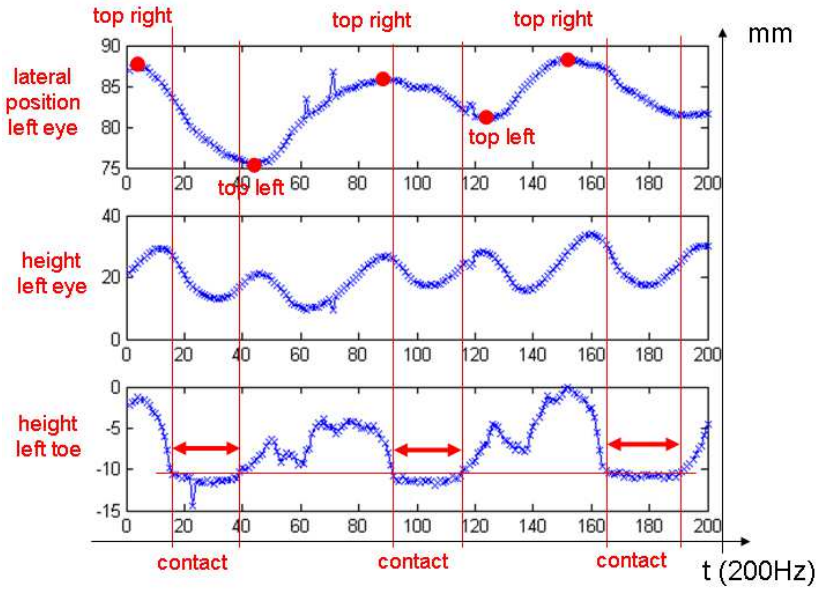


Figure 2.13: *Example of 3D trajectories of markers during rat locomotion.*

2.3 ANIMATION MODEL OF THE INTERNAL STRUCTURES

So far, the models that can be animated are simply geometry : a set of vertices and faces with no articulation between the different meshes representing the different skeletal structures. To animate such models, granted that the 3D trajectories of points on the models are known, one can resort to SVD-based methods [BBG+10] where the rigid motion of a bone is computed as the rigid transformation of a set of markers implanted on that bone. However, such a method requires at least 3 markers per bone to animate, which is very hard to achieve for motion capture of the complete skeleton of small animals. To be able to loosen the experimental constraints by reducing the number of markers required, one must give an underlying structure to this set of independent meshes.

2.3.1 Set of articulated solids

To create an underlying structure to the set of meshes, the skeleton is modeled as a set of articulated solids. Each solid represents one bone or a group of bones. The bone or bones represented by a solid will be called the *anatomical entity* of the solid. Each solid has 6 degrees of freedom (3 in translation and 3 in rotation) expressed in the world reference frame. Given one anatomical entity, the mesh or meshes modeling the entity are rigidly mapped to the solid representing it. As a result, every rigid transformation applied to the solid will be applied to the mesh(es). The solid can be considered as the *animation model* of the anatomical entity and the meshes as the *visual model* of the anatomical entity. For example, in Figure 2.14, the forearm is an anatomical entity represented by one solid that rigidly drives a visual model made of two meshes : one modeling the radius and one modeling the ulna. The same way, the humerus is an anatomical entity represented by one solid that rigidly drives a visual model made of only one mesh modeling the humerus.

To enforce anatomical constraints on the anatomical entities, we define articulations between pairs of solids. They are illustrated on Figure 2.14. An articulation between two solids is modeled by creating a reference frame at the position of the articulation for each solid. Those reference frames are called *joint frames*. They are defined by their position (3 degrees of freedom) and their orientation (3 degrees of freedom). Those degrees of freedom are rigidly mapped to the solid they belong to. As a result, at each articulation between two solids, there are two joint frames, each rigidly mapped to one of the two solids. Figure 2.14 shows a 3D representation of those joint frames at the top right and a 2D representation at the bottom (red).

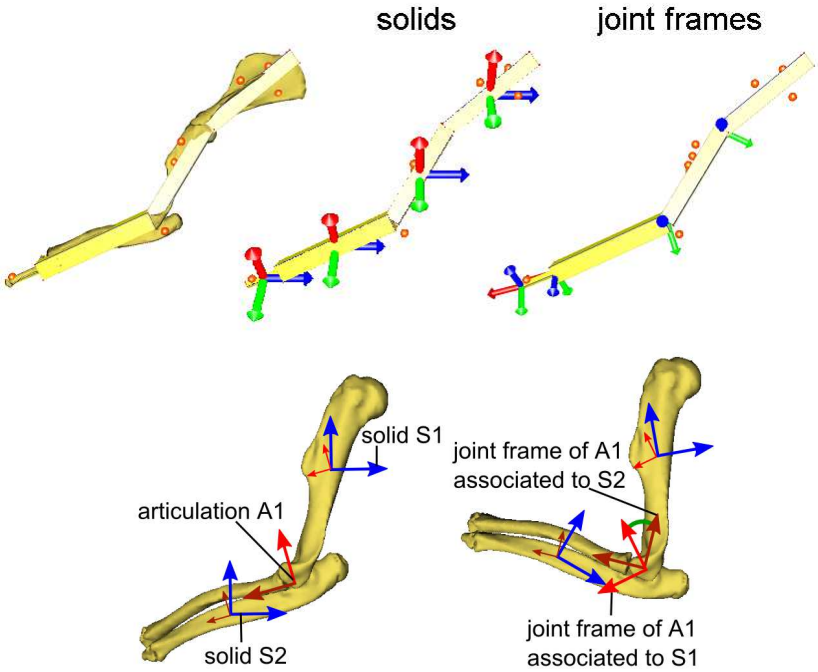


Figure 2.14: *Animation model - top : Bone models with visualisation of solid bodies (left), frames defining the solid bodies (middle), joint frames (right) - bottom : Each solid body (blue) is rigidly connected to a joint frame (red and dark red). On the rest pose (left), the joint frames are the same. Once the bones move with respect to each other (right), each joint frame stays rigidly connected to one of the body (see small red frames), resulting in different transformations for the two joint frames (green).*

On the rest pose, the pose in which the solids and articulations are defined, the two joint frames at an articulation are defined to be the same i.e. they have the same position and orientation. As the two joint frames are rigidly mapped to different solids, when the solids move with respect to one another, the joint frames are subjected to different rigid transformations. Therefore, in a deformed pose, the deformation induces a translation and/or a rotation in-between the two joint frames, see bottom right of Figure 2.14. For any given deformed pose, there are therefore two types of measures of motion available. First, the position and orientation of all anatomical entities are known in the world reference frame. Second, the transformation from the rest pose of one solid with respect to the other solids it shares an articulation with can be computed in the solid reference fame from the joint frames.

The level of details of the modeling depends on the level of details of the anatomical entities. For instance, each carpal bone can be considered one anatomical entity or, on the other end of the scale, all carpal bones as well as the metacarpal bones and the

phalanxes can be grouped in one anatomical entity : the front paw. Figure 2.15 show two different levels of details that can be considered for the mouse. Treating each separate bone individually is a thorough approach but increases the complexity of the problem, sometimes for irrelevant results, such as each individual carpal bone. On the other hand, if restricting the number of anatomical entities to the few articulations of interest seems interesting, experience shows that it is often hard to choose beforehand which articulations will turn up to be of interest. For instance, the spine is made up of 20 vertebrae, when excluding the tail. Knowing beforehand which articulations to model in-between vertebrae is hard to predict as the movement is usually distributed over all vertebrae. As a result, a trade-off between the complexity of the problem and the precision of the model must be found depending on the final goal of the tracking. In this thesis, we propose a model that is suited for studies of the global movement of rodents, not a particular articulation. For instance, our model is suited to measure the movement between the femur and the tibia, but not to measure the movement of the patella with respect to those two bones.

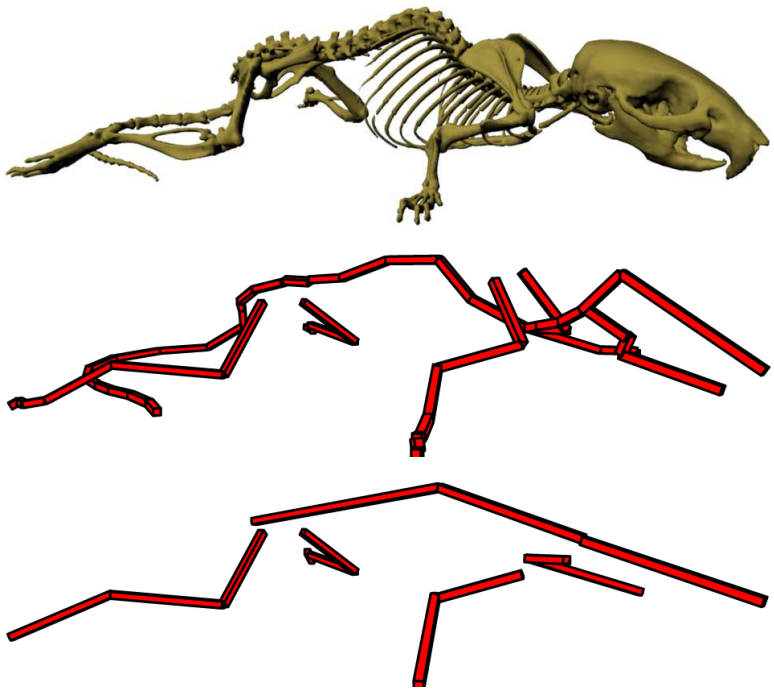


Figure 2.15: *Different level of details possible to model a mouse. The solids are displayed as red parallelepipeds.*

As a result, we model rodent morphology as illustrated on Figure 2.16 and as follows :

1. The head is modeled as one solid.

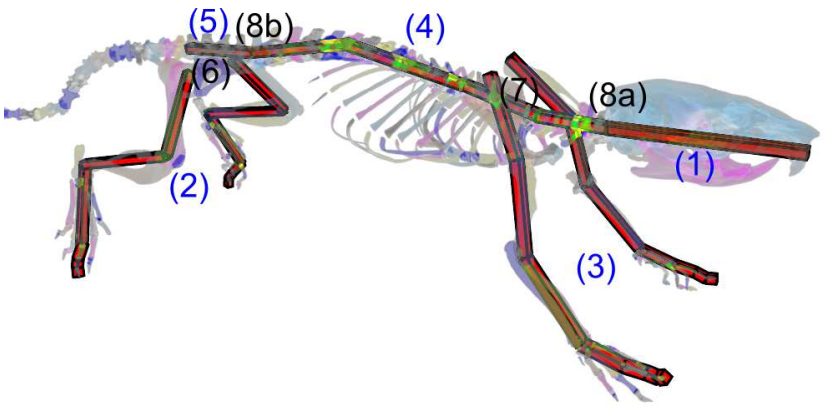


Figure 2.16: *Animation model for rodents. The solids are displayed as red parallelepipeds.*

2. The hind legs are modeled with one solid for the femur, one for the tibia, one for the tarsal and metatarsal bones, one for all the proximal phalanxes, one for all the middle phalanxes and one from all the distal phalanxes, with one articulation between two successive solids.

3. The forelegs are modeled with one solid for the scapula, one for the humerus, one for the radius and ulna, one for the carpal and metacarpal bones, one for all the proximal phalanxes, one for all the middle phalanxes and one all from the distal phalanxes, with one articulation between two successive solids.

4. The spine is modeled by four solids : one between the first cervical vertebra (C1) and the seventh one (C7), one between C7 and the sixth thoracic vertebra (T6), one between T6 and the first lumbar vertebra (L1) and one between L1 and the sixth lumbar vertebra (L6), with articulations in-between two successive solids.

5. The pelvis is modeled as one solid.

6. One articulation is created at the head of the femur between the pelvis and the femur.

7. Two articulations are created at C7 between the scapulas and the block C7-T6.

8. The spine is articulated to the rest of the body through two articulations : one at C1 between the head and the C1-C7 block (8a) and one at L6 between the pelvis and the L1-L6 block (8b).

As a result, we have a set of articulated solids defining anatomical entities. Therefore, each individual bone can be described in the animation model by the anatomical entity to which it belongs. This entity has an associated solid defined entirely by its position and orientation. Given the anatomical entity to which it belongs, the motion of each bone is thus defined by the motion of the associated solid.

2.3.2 Refined model of the spine and ribcage

The model described in Section 2.3.1 only has four solids to describe the motion of the spine and the ribcage attached to it. For large deformations, this can lead to a 'blocky' aspect as can be seen on Figure 2.17. On top of visual aspects, such a small number of solids prevents a precise measure of the position and orientation of each vertebra. On the other hand, modeling all vertebrae leads to a complex model that is not necessarily required. We propose a trade-off between accuracy and complexity by modeling the spine and ribcage through a 3D cubic Hermite spline driven by the four solids describing the spine.

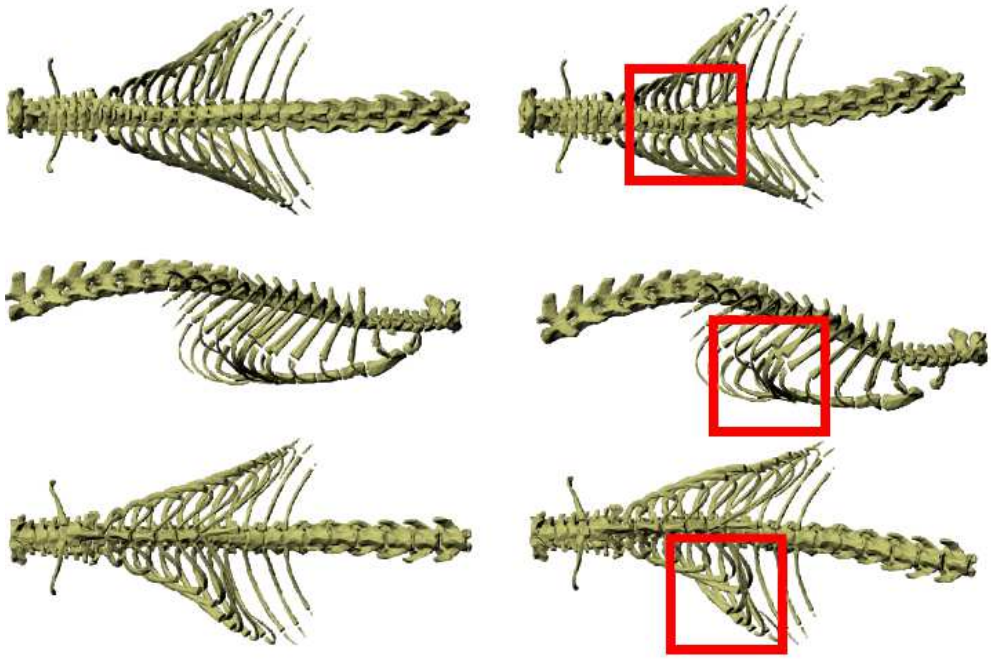


Figure 2.17: *Top, side and bottom view of the spine and ribcage in the rest pose (left). When deforming the spine and ribcage using only 4 solids (right), the ribs separate (bottom), inter-penetrate (middle) and the deformation of the vertebrae is not smooth (top).*

Hermite splines are used to smoothly interpolate between key-points. An Hermite spline is a third-degree spline, defined by two control points and two control tangents. It is very easy and fast to calculate. To interpolate a series of points P_k, $k = 1..n$, interpolation is performed on each interval (P_k, P_{k+1}). The equation of the Hermite spline between P_k and P_{k+1} is :

$$C_k(u) = h_{00}(u)P_k + h_{10}(u)D_k + h_{01}(u)P_{k+1} + h_{11}(u)D_{k+1} \tag{2.1}$$

with $u \in [0, 1]$, D_j is the tangent at control point P_j and h_{ij} are the basis functions and defined as :

$$\begin{aligned} h_{00}(u) &= 1 - 3u^2 + 2u^3 \\ h_{10}(u) &= u - 2u^2 + u^3 \\ h_{01}(u) &= 3u^2 - 2u^3 \\ h_{11}(u) &= -u^2 + u^3 \end{aligned} \tag{2.2}$$

Figure 2.18 shows an example of a 2D Hermite spline.

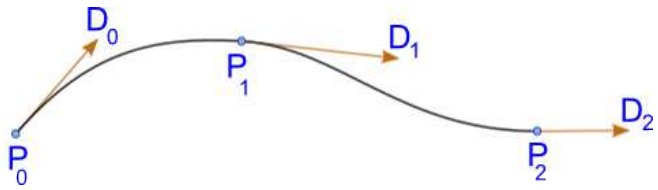


Figure 2.18: *Example of a 2D Hermite spline with 3 control points.*

We model the spine and ribcage with an Hermite spline. A rodent has 7 cervical vertebrae, 13 thoracic vertebrae and 6 lumbar vertebrae. The ribcage is made up of the ribs and the sternum (see Figure 2.19). Each thoracic vertebra has a rib attached to it. The first seven ribs are attached to the sternum that is composed of 6 bones.

The P_k of Equation (2.1) are chosen as the mid-points between the two joint frames at each articulation between the four solids defining the spine. Figure 2.20 illustrates the process on two solids. This gives us five points. Their tangents, i.e. the D_k of Equation (2.1), are computed by finite difference as :

$$D_k = \frac{P_{k+1} - P_k}{2} + \frac{P_k - P_{k-1}}{2} \tag{2.3}$$

We are therefore able to compute the Hermite spline on the rest pose. For each bone i composing the spine and ribcage, the barycenter B_i of the mesh is computed. The point P on the Hermite spline that is closest to B_i is stored as two components : the segment k_i it belongs to and the parametric coordinate u_i such that $P = C_{k_i}(u_i)$. We also store P as the initial position on the Hermite curve. Figure 2.21 shows the Hermite spline reconstructed on the rest pose. Explanations on how the Hermite spline

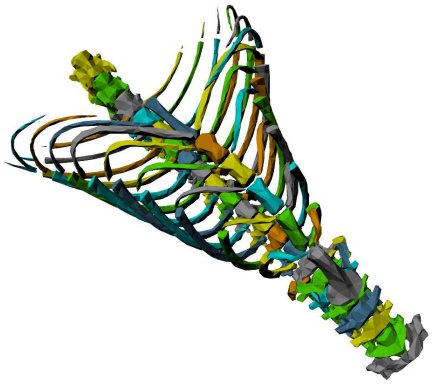


Figure 2.19: *Segmentation of the different bones constituting the ribcage.*

is used during animation and how it improves the animation of the spine and ribcage with respect to the solids' animation are shown in Section 2.4.3.

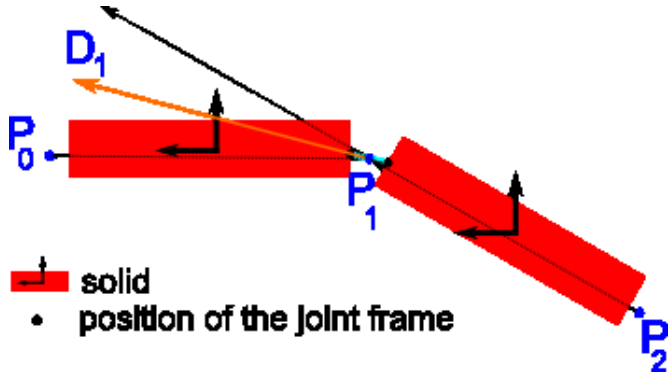


Figure 2.20: *Illustration of how the control points and tangents of the Hermite spline are computed from the set of articulated solids.*

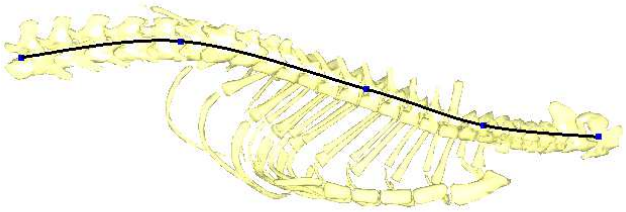


Figure 2.21: *3D Hermite spline fit on the rest pose of the animation model.*

In the end, each individual bone segmented from the CT-scan has its motion defined in either one of these ways :

- it is defined by the anatomical entity it belongs to and its motion is the same as the corresponding solid
- it is defined by a segment, a parametric coordinate and an initial position on a Hermite spline and its motion will be described in Section 2.4.3.

2.4 MARKER-BASED TRACKING OF THE INTERNAL STRUCTURES

We now have an animation model where each piece of geometry defining internal structures is defined either by their corresponding solid or by a parametric coordinate on a Hermite spline. The Hermite spline itself is computed from the positions of the 4 solids of the spine. The problem of tracking can therefore be formulated as : how do we set the degrees of freedom of the solids so that the visual models fit the videos? The motion cues extracted from the videos are the 3D trajectories of the markers (Section 2.2.2).

2.4.1 Sequence-specific model

In Section 2.3, we have developed a generic rodent animation model. However, every animal that is captured has a different morphology. On top of that, for every sequence, a different set of markers is implanted. This leads us to develop a sequence-specific model for each sequence of motion capture from the generic animation model. The specificity comes in two forms : the size of the animal and the positions of the markers.

Definitions of landmarks on the animation model

To be able to register the animation model to the markers, one must first know where the markers are on the animation model. To do so, we create what we call *landmarks* on anatomical entities. These landmarks are the positions of the markers with respect to the anatomical entities in the rest pose. They are rigidly mapped to an anatomical entity. Figure 2.22 shows how landmarks are submitted to the rigid transformation of the solid they are associated with.

In the best case scenario, the animal used in the study has been scanned with the markers implanted as in [BBG+10]. This way, the position of the landmarks can directly be measured from the CT-scan. However, as mentioned previously, this is often not the case. To overcome this issue, several strategies can be used, differing in the amount of user intervention involved. First of all, these positions can be specified manually. This can be done with the visual help of the model superimposed on the videos. This way, if the model is manually rotoscoped as in [GBJD10], the 3D position of the marker as computed in Section 2.2.2 can be used as landmark. To automate the process, if the marker is implanted on the bone surface, once the animation model is rotoscoped, the point on the bone model that is closest to the 3D position of the marker can be selected as landmark.

Most of the time, however, the makers are glued on the skin that is not rigidly connected to the bones. This makes the assumption that a landmark is rigidly mapped to an anatomical entity false (see the white marker on the tibia on Figure 2.2). The position of the landmark is therefore an approximation, not a ground truth that can be

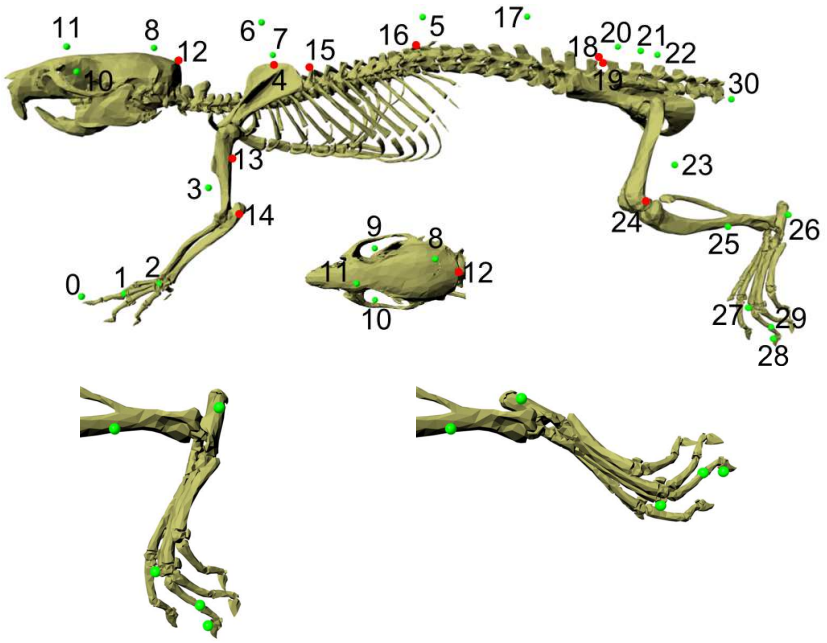


Figure 2.22: *Landmarks (green and red spheres) are rigidly mapped to anatomical entities. The green landmarks correspond to external markers, the red landmarks correspond to internal markers, not visible on standard cameras. When the anatomical entity moves (bottom), the landmarks are submitted to the same rigid transformation.*

measured by a scan. To better approximate this position, we develop a trial-and-error method during tracking that will be explained in Section 2.4.4.

Morphological adaptation to a specific subject

Most of the time, the CT-scan of the animal that is captured through videos is not available. The generic rodent animation model developed in Section 2.3 has to be adapted to fit the animal being captured. To do this, we align features that can be measured both on the generic model and the videos. In our case, we align a set of distances between landmarks that are rigidly connected to the same distances between the corresponding markers. More precisely, the measured features are the distance between the eyes, the distance from the front to the back of the head, the length of the forearm and the length of the hindleg (see Table 2.1).

Indeed, from the 3D trajectories of a given pair of markers, we can compute the 3D distance between those two markers through the n frames of the sequence. From this series of distances, one can extract the mean and standard deviation of the distance and compute the 95% confidence interval of the distance. For a confidence interval smaller than 1mm, we can approximate the parameters with the mean.

The distance between the two markers can also be computed on the animation model by the distance between the landmarks. From the ratios between those two measures for the different pairs of markers, we can once again compute the mean, standard deviation and 95% confidence interval of a global scale to fit the animation model to the animal being captured.

For sequence ratC6X1M1lo whose set of markers can be seen on Figure 2.22, Table 2.1 shows the mean, 95% confidence interval and ratio of the distance measured on the tracking over the distance measured on the landmarks. One can see that the assumption of rigidity is not unrealistic as the confidence interval for $n = 300$ frames is at most 1.34mm large, which corresponds to 5% of the estimated distance. The mean of the different ratios is 0.75 with a 95% confidence interval of $[0.71, 0.78]$. This seems like quite a large confidence interval. However, using either 0.71 or 0.78 as scaling factors leads to a difference in the estimation of the distance of at most 2.16mm, which is 8% of the estimated distance. We therefore use a global scale for the whole model.

pair of markers	estimated distance (mm)	95% confidence interval (mm)	distance of the landmarks (mm)	ratio
09-10 (eyes)	17.10	$[17.08, 17.13]$	23.39	0.73
08-11 (head's length)	23.39	$[23.35, 23.42]$	29.08	0.80
02-03 (forearm)	26.91	$[26.33, 27.49]$	37.97	0.71
23-25 (hindleg)	25.47	$[24.80, 26.14]$	34.49	0.74

Table 2.1: *For n = 300 frames, mean 3D distance measured between pairs of tracked markers as well as the 95% confidence interval, same distance measured on the landmarks and ratio of the mean distance over the distance of the landmarks. See Figure 2.22 for the location of the markers.*

Thanks to the definitions of landmarks on the model and the morphological adaptation, we can adapt the generic animation model of Section 2.3 to the sequence being studied, both in terms of the size of the animal and the set of markers implanted.

2.4.2 Inverse Kinematics

Once the 3D trajectories of the markers are computed and the landmarks are located, applying inverse kinematics [ZB94] enables us to solve for the degrees of freedom of the r solids with the objective to put the landmarks at the positions of the tracked markers. The positions of the landmarks are computed by applying the rigid transformation of the solid to which they are mapped to their positions in the rest pose. The 3D position of a landmark x_i who is associated to a solid with degrees of freedom q_j can be written as :

$$x_i = g(q_j) \tag{2.4}$$

where g is the result of applying the rigid transformation described by q_j to the landmark's initial position. This is *forward kinematics*. *Inverse kinematics* (IK) consists in setting the position x_i of the landmark and search for q_j such that Equation (2.4) is true. Equation (2.4) can be written for all m landmarks and lead to equation :

$$x = f(q) \tag{2.5}$$

where $x = (x_i)_{i=1..m}$, $q = (q_i)_{i=1..r}$ and $f(q) = (g(q_{j_i}))_{i=1..m}$ where j_i is the solid that drives the landmark i.

As f is not linear, the problem can have zero, one or many solutions though a closed-form solution can be derived in particular cases. To solve the IK problem, it is standard to linearize Equation 2.5 around the current configuration :

$$f(q + \delta q) = f(q) + J(q)\delta q \tag{2.6}$$

which leads to

$$\delta x = f(q + \delta q) - f(q) = J(q)\delta q \tag{2.7}$$

where J is the jacobian matrix of f i.e.

$$J_{ij} = \frac{\partial f_i}{\partial q_j} \tag{2.8}$$

For a small displacement, solving Equation (2.5) comes down to computing the inverse of J to compute the variations of the degrees of freedom δq.

In general, J is non-square, non-inversible. In those cases, the pseudo-inverse of J can be used :

$$\delta q = J^+(q)\delta x \tag{2.9}$$

If $m < n$, the system is under-determined, there are many solutions. The pseudo-inverse is

$$J^+ = J^T(JJ^T)^{-1} \tag{2.10}$$

and leads to the solution that minimizes the euclidean norm of δq i.e. chooses the solution that creates the less variation in the degrees of freedom. If $m > n$, the system is over-determined and there may be no solution. In that case, the pseudo-inverse of J is

$$J^+ = (J^T J)^{-1} J^T \tag{2.11}$$

and leads to the solution in the least-square sense, the one that minimises

$$\sum_i \|\hat{x}_i - x_i^*\|^2 \tag{2.12}$$

where $\hat{x}_i$ is the achieved position of landmark i and x_i^* is the tracked position of marker i.

As we use a linearisation of the IK equation, this solution is valid for small displacements. To have the solution Δq such that

$$x^* = f^{-1}(q^0 + \Delta q) \tag{2.13}$$

where q^0 is the initial value of q and $x^* = (x_i^*)_{i=1..m}$ we iterate on small displacements until we reach x^*.

However, IK has a few drawbacks. It involves inverting a matrix, which can be numerically unstable. On top of that, all the markers have the same influence. However, as the external markers are not rigidly coupled to the skeletal structures, some markers are more or less representative of the position of the anatomical entity they are associated with. For example, the assumption of rigidity between the marker and the bone is much more accurate for a point at the ankle where there is not a lot of flesh than for a point on the tibia (see white point in Figure 2.2). As a result, to find the degrees of freedom, we have much more confidence on the marker at the ankle than on the marker at the tibia. Thus, if the landmark at the tibia does not reach the position of the marker, it is less likely that this means the tibia is badly registered than if the landmark at the ankle does not reach the marker. It may simply be because the rigidity constraint is not valid. Finally, to integrate anatomical constraints at the articulations through the joint frames, one must either solve the IK system with constraints on q or has to reparameterise the animation model with reduced degrees of freedom to have a hierarchy of joints whose degrees of freedom are the degrees of freedom at the articulations. We will see later on how such a parameterisation is not suitable for rodents.

For these reasons, we formulate the tracking as a weighted Inverse Kinematics problem modeled as a forward dynamics system. Indeed, another way of modeling the IK problem is to consider that markers attract landmarks the way a spring would [ZVDH03]. Such a spring creates a force f_x applied on the landmarks. As the landmark is rigidly mapped to a certain solid, this force is mapped on the solids as a force f_q. According to the principle of virtual work :

$$\dot{q}^T f_q = \dot{x}^T f_x \tag{2.14}$$

where $\dot{b}$ is the derivative of b with respect to time. We have seen that for a small displacement, linearising gives us Equation (2.7). Substituting Equation (2.7) in Equation (2.14) gives :

$$f_q = J(q)^T f_x \tag{2.15}$$

Applying Newton's second law, we can compute $\ddot{q}$, the second derivative of q, from

$$M\ddot{q} = f_q \tag{2.16}$$

using the conjugate gradient method for instance. We can then integrate $\ddot{q}$ to compute $\dot{q}$, then q. This moves the landmarks towards the positions of the markers x. The

equilibrium of the dynamic system is reached when f_q is minimized i.e. when $J^T f_x$ is minimized. As f_x is the spring force between the landmarks and the markers, the system is stable when the distance between the markers and the landmarks is minimized in the least square sense. Once q is computed, the position and orientation of the visual models can be computed by applying, for $i = 1..r$, the rigid transformation of q_i to the meshes that are rigidly mapped to solid i.

In this formulation, constraints at the articulations are easily modeled as angular springs between the two joint frames at the articulations (see Section 2.3.1). These springs try to preserve contact between the two bones (with the translation part of the spring) and to get back to the rest pose (with the orientation part of the spring). On the bottom row of Figure 2.14, angular springs would try to prevent the transformation (green) between the two joint frames (red). This is done by adding a second force f_{jt} to the system that accounts for the angular springs. As with the landmarks, the rigid mapping between the joint frames and the solids leads to f_{jt} being mapped to f_q by the transpose of the jacobian linking small variations of q to small displacements at the articulations. What is minimized when the system is stable is therefore :

$$J_x^T(q) f_x + J_{jt}^T(q) f_{jt} \tag{2.17}$$

The first part of the sum can be considered as a tracking constraint : the linear springs penalise the distance between landmarks and markers. On Figure 2.23, you can see the model in its rest pose and the linear springs that are going to be applied to the model. The second part can be considered as an anatomical constraint : the angular springs penalise solids moving away from one another and from the rest pose configuration, which is known to be a valid natural pose.

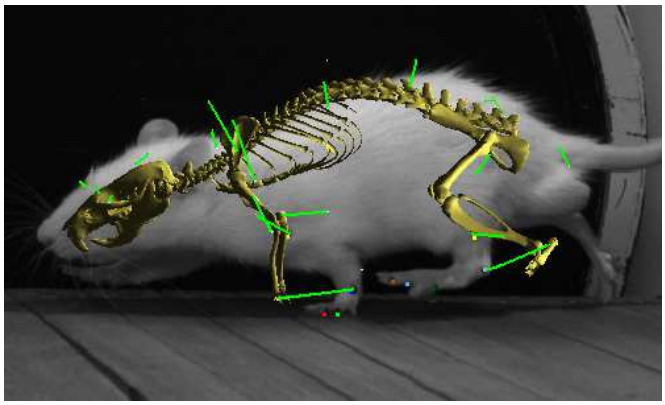


Figure 2.23: *Examples of linear springs (green lines) between markers and landmarks for the set of markers of Figure 2.22 when the skeleton is not yet aligned.*

The influence of those different penalties can be set through the stiffnesses of the linear and angular springs. Indeed, the higher the stiffness, the higher the penalty. The more likely the assumption of rigidity between a marker and a solid is to be valid, the higher the stiffness of the associated linear spring. As for the angular springs, there are two stiffnesses : one in translation and one in rotation. The stiffness in translation is usually set high with respect to the stiffness of the linear springs and to the stiffness in rotation because except between the scapula and the the spine, it it unlikely that two articulated solids separate. Figure 2.24 illustrates the sliding movement the scapula has with respect to C7. This also validates the choice not to use reduced degrees of freedom by modeling the animal as a hierarchy of joints. In this type of model, the degrees of freedom are the rotations at the articulation and do not allow for leniency in translation, which makes modeling the movements of the scapula harder.

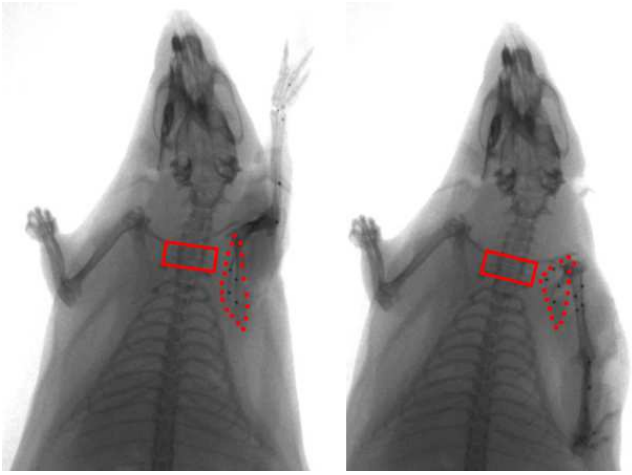


Figure 2.24: *The movement of the scapula (delimited by red dots) with respect to the spine (C7 is enclosed in the red square) can not be considered to be a pure rotation, a translation has to be modeled as well.*

To better understand the influence of adding angular springs, we compare three methods to register the bones to the markers :

1. [BBG+10], an SVD-based method where the rigid transformation of at least 3 markers is computed at each instant and applied to the geometry,
2. our method **without** the angular springs,
3. our method **with** the angular springs.

Figure 2.25 shows the results on the tracking of the humerus and the forearm (radius and ulna) of a rat with 4 markers on each anatomical entity. The radius-ulna complex is a long thin entity. Placing 3 non-collinear markers on such structures is difficult. As a result,computing rigid transformations between those 'barely' non-colinear markers

is close to ill-conditioned. This can lead to an unnatural roll of the radius with respect to the humerus when using SVD-based methods. Not using the angular springs in our method produces the same artefacts than with an SVD-based methods. Adding angular springs regularises the solution and prevents the roll of the forearm.

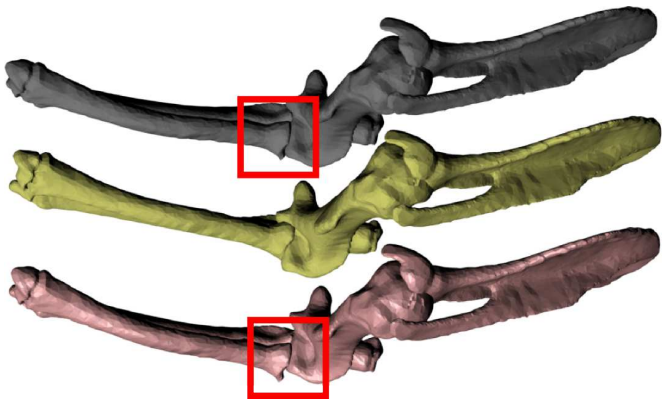


Figure 2.25: *From top to bottom : tracking of the forelimb of the rat using an SVD-based method (grey), using our method with active joint springs (yellow), using our method with inactive joint springs i.e. with a null stiffness both in rotation and in translation (pink). Notice how the joint springs correct an invalid roll of the radius.*

The influence of the stiffness in rotation can also be seen in Figure 2.40 where we can see that lowering this stiffness leads to a better reconstruction. Indeed, if a marker is directly implanted on the bone (hence there is a perfect rigid connection between the landmark and the bone) then we can trust this marker to be a good clue to locate the bone and lower the influence of the rest pose by lowering the stiffness in rotation. This can also be seen on Figure 2.26.

Figure 2.27 shows how using different stiffnesses for the different markers can also improve tracking. Indeed, if the assumption of rigidity is ill-suited (as for marker 3 on Figure 2.22) then lowering the stiffness of that marker leads to a better registration as the angular springs kick in to help find a correct pose.

In practice, the stiffnesses of the angular springs are set to be the same for all angular springs with the stiffness in translation three times larger than the stiffness in rotation. The stiffnesses of the linear springs are set individually for each marker depending on the confidence in the rigidity hypothesis.

Thanks to weighted Inverse Kinematics, we have recovered the positions and orientations of the solids. Through rigid transformation, this gives us the position and orientation of the different bones. However, as explained in Section 2.3.2, due to coarse-

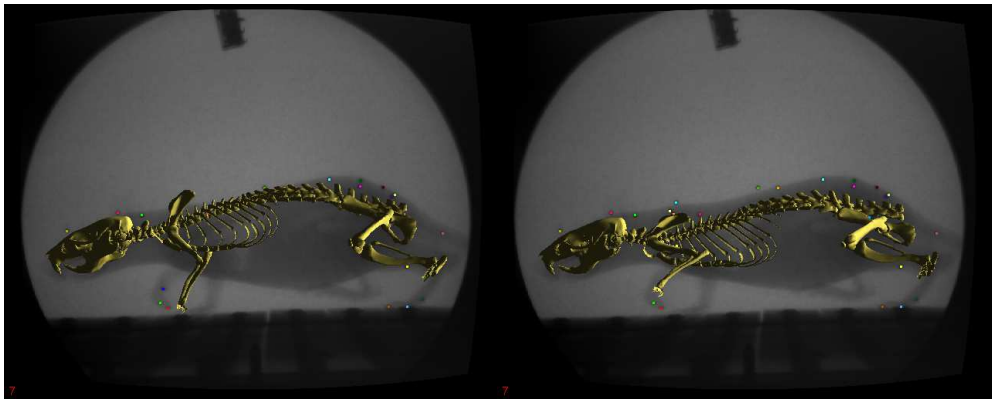


Figure 2.26: *Choosing a high stiffness in rotation for the angular springs in comparison to the stiffnesses of the linear springs (left) leads to a solution closer to the rest pose but further away from the markers. Decreasing this stiffness leads to results closer to the markers (right).*

ness of the model for the spine, the vertebrae and ribcage deform in an unnatural way (Figure 2.17).

2.4.3 Update of the spine and ribcage

To prevent the 'blocky' deformation of the spine and ribcage due to the approximation of the spine by only four solids, we modeled the individual bones of the vertebrae, ribs and sternum through a Hermite spline. This Hermite spline is built from the articulations in-between the solids of the spine in the articulated model. Once the IK system is solved as in Section 2.4.2, we can compute the new joint frames at each articulation through rigid transformation as each joint frame is rigidly driven by one solid. Once the mid-points of the articulations are computed, the Hermite spline can be updated following Equation (2.1).

A bone of the spine and ribcage is defined by a segment k, a parametric coordinate u and an initial position P. The translation that is applied to the bone is therefore computed as :

$$T = C_k(u) - P \tag{2.18}$$

The rotation to apply to the bone is computed by spherical linear interpolation (slerp) between the rotation of the neighbouring solids. Indeed each segment k of the Hermite spline corresponds to a solid. The rotation of each bone belonging to the solid k is computed as an interpolation of the the rotation of the solid $k-1$ and the rotation of the solid $k+1$ with the interpolating coefficient u. Figure 2.28 shows an example on how the use of the Hermite spline improves the tracking of the spine and ribcage.

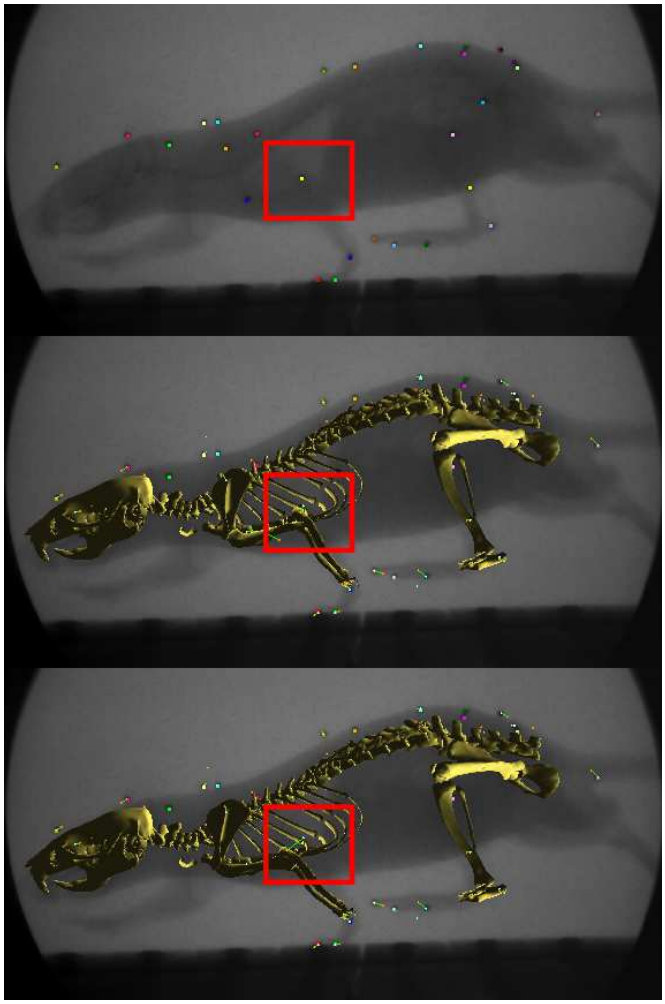


Figure 2.27: *From top to bottom : frame of an experience with the set of markers of Figure 2.22 - tracking with a stiffness on marker 3 on the forearm lower than other stiffnesses - tracking with a stiffness on marker 3 twice the stiffness on other markers.*

Thanks to Inverse Kinematics and Hermite spline interpolation, we can compute the position and orientation of each bone of the skeleton. However, this tracking relies heavily on the location of the landmarks with respect to the bones. As explained in Section 2.4.1, this process is usually manual and sometimes difficult. We have however developed a strategy to improve this estimation.

Figure 2.28: *To correct the artefacts created by having only 4 solids to model the spine (left), we use Hermite spline interpolation and spherical linear interpolation to set the position and orientation of the ribcage (left).*

2.4.4 Automatic refinement of the location of the landmarks

Manual placement of landmarks is highly unreliable depending on the number of cameras and on the actual placement of the marker. Indeed, a marker implanted on an anatomical point such as the wrist or ankle is easier to locate than a marker implanted at a place where there is a lot of soft tissues, on the thigh for instance. Unfortunately, an error in the location of the landmarks can completely spoil tracking results. To compensate for this limitation, we have developed a trial-and-error strategy. Once a tracking solution is computed, improving the location of the markers can be done by measuring the elongation of the linear springs.

Indeed, after a first estimation of the animation, if a linear spring produces a consistent elongation vector between the landmark and the tracked 3D position in the corresponding solid's reference frame throughout the sequence, one can assume that the landmark placement is slightly off. This can be corrected by shifting the landmark by this elongation vector.

On all the sequences from the University of Brown described in Section 2.5.1 where the animal and set of markers are the same, after a manual estimation of the position of the landmarks, we track the foreleg of the rat through 736 frames. We applied the automatic refinement method from this tracking for the 4 markers on the scapula, the 4 on the humerus and the 4 on the radius-ulna. We then tracked again the foreleg with the new positions of the landmarks. Table 2.2 shows that applying this correction significantly lowers the elongations of the linear springs as the mean elongation is 0.6597mm without correction and 0.2688mm after applying the correction.

We have therefore developed a tracking method that registers a generic rodent animation model to a set of 3D trajectories of markers. To do so, landmarks are first defined

		Scapula	Humerus	Radius-Ulna
Without correction	mean	0.3618	0.6088	1.0086
	s.d.	0.1321	0.1696	0.1480
	max	0.9146	1.4286	2.7222
With correction	mean	0.1232	0.2615	0.2961
	s.d.	0.0896	0.1035	0.1232
	max	0.2488	1.0200	1.1161

Table 2.2: *Distances in mm between landmarks and tracked 3D positions before and after automatic refinement of the location of the landmarks for* $n = 736$ *frames.*

on the model and the animation model is then scaled down to fit the captured animal. Once the sequence-specific animation model is created, the positions and orientations of the different skeletal structures are tracked through weighted Inverse Kinematics and refined through Hermite spline interpolation.

2.5 VALIDATION OF THE ANIMATION MODEL

To evaluate the animation model of the internal structures and the use of weighted IK to animate it, we used video-radiography to be able to see the true positions and orientations of the internal structures.

2.5.1 Case study : motion of the foreleg of the rat during locomotion

To be able to quantitatively, and not only qualitatively, evaluate the different steps of the marker-based tracking of the internal structures, we use the bi-planar X-ray videos of sequence ratC0X2M1lo to be able to measure internal structures in 3D. The animation model is therefore composed of 3 anatomical entities : the scapula, the humerus and the radius-ulna (see Figure 2.29). The same animal was not used for the scan and for the data recording. As a result, the model had to be scaled down and the landmarks estimated. The landmarks were placed using automatic image-based registration on one manually segmented frame and selection of the point on the bone model closest to the marker. As explained previously, after a first estimation of the tracking, the position of the landmarks were automatically refined.

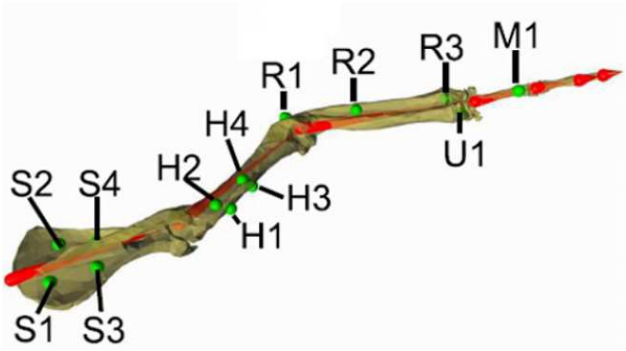


Figure 2.29: *Model for ratC0X2M1lo : bone models (yellow) with solids (red) and markers (green) added to the model.*

Validation of the calibration

The calibration objects used to calibrate the X-rays are described in [BBG+10]. We assess the accuracy and precision of both the distortion correction and of the camera calibration. The precision of the distortion correction is measured as the collinearity of the steel beads on the acrylic sheets (Figure 2.30). For each line of beads, a linear regression is performed to compute the best 2D line fitting the projections of the beads on the cameras. The deviation from collinearity is measured as the distance to this line

of the projected beads. The beads project as discs of about 10 pixels of radius. Before distortion correction, the mean of those distances is 3.9±5.1 pixels (see Figure 2.31 for their distribution). As Figure 2.31 illustrates, once the distortion of the images is corrected, most of those distances are below 2 pixels. This results in a mean of 2.0±2.6 pixels. Distortion has a greater impact the farther away from the image center (see Figure 1.7). As a result, correcting the distortion mostly corrects the position of the beads projecting on the border of the images. This is why we see that the distribution is far less spread and the maximum distance is smaller when distortion correction is applied. Distortion correction brings back the beads that were creating the greatest distances (up to 24 pixels) to a more correct position.

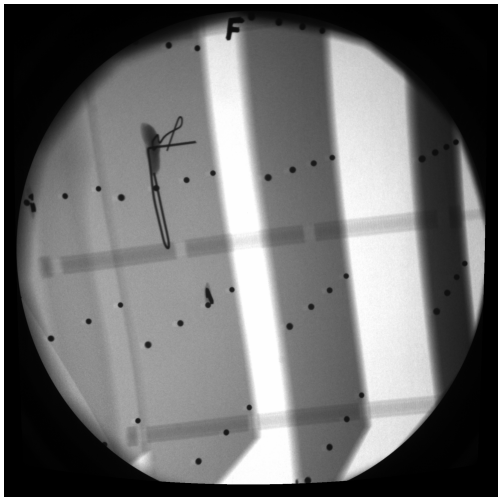


Figure 2.30: *Calibration object used for bi-planar X-ray sequences.*

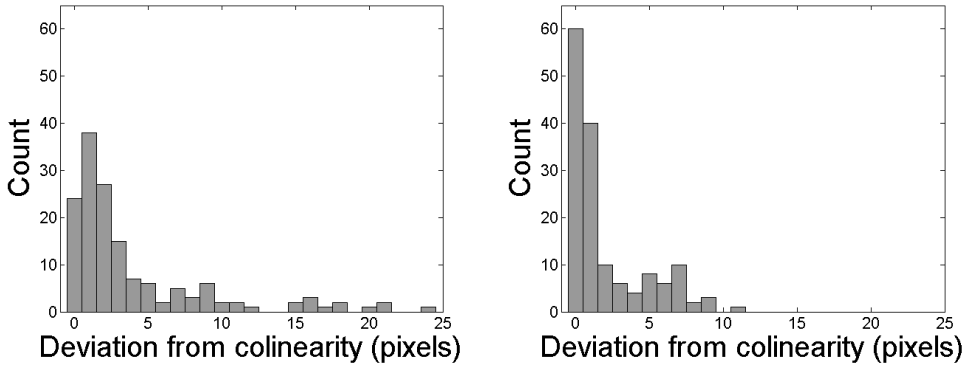


Figure 2.31: *Distribution of the collinearity error before (left) and after (right) distortion correction.*

Once the cameras were calibrated, the calibration object of Figure 2.30 was reconstructed and the distances between the different steel beads were computed and compared to ground truth. Ideally, this validation should have been done on a different object than the one used to calibrate. Unfortunately, this was the only object of known measures that was recorded. This object was manufactured so that the beads are spaced every 65mm in each of the 3 directions. Once reconstructed, the mean distance found between the reconstructed 3D position was 64.8805mm with a standard deviation of 0.9619mm. As the error is below 0.5mm, the calibration can be considered correct and as the standard deviation is below 1mm, it can be considered precise.

Validation of the 3D trajectories of the markers

Once the calibration is validated, we can assess the accuracy and precision of the tracking of the markers. Indeed, when tracking the points in 2D, errors can occur. They may come from automatically tracking the markers or from marker ambiguity on small structures (see for example the markers on the humerus on the dorso-ventral view of Figure 2.32). One way to diagnose those errors automatically is to compute the distance between the tracked 2D points and the epipolar lines derived from the other view. Indeed, this distance should be null if the tracking is perfect and an error in 2D tracking will create outliers. Plotting the distances with respect to time can highlight those outliers that can then be corrected (see top row of Figure 2.33). On the 736 frames on the bi-planar X-ray, the mean distance to the epipolar lines is 0.5618 pixels, which is negligile.

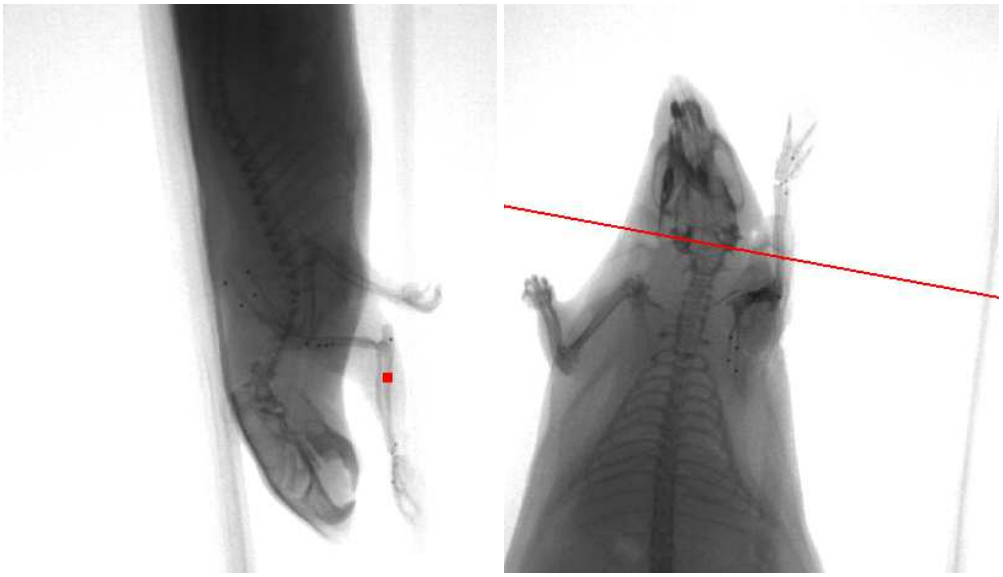


Figure 2.32: *Tracked point in 2D (red point on the left image) and corresponding epipolar line (red line on the right image). The marker should project somewhere on the epipolar line on the right image.*

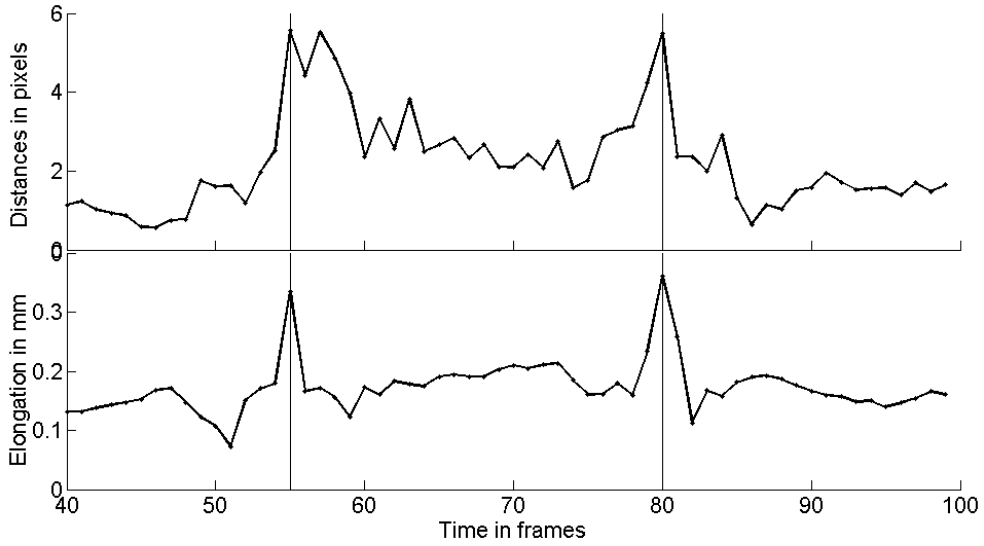


Figure 2.33: *Diagnosis of errors in the tracking of markers - top row : sum of the distances between the tracked 2D points and the epipolar lines for the markers attached to the radius, bottom row : elongation created by the markers on the radius.*

As the markers are implanted directly on the bones, one can measure the accuracy of the 3D tracking by measuring the distance between markers implanted on the same bone and checking that this distance stays constant through the tracking. Mathematically, this means that the standard deviation of the distance between markers implanted on the same bone measured over the 736 frames should be null. We computed the distance between the 18 pairs of rigidly-connected markers of our experiment on the 736 frames available. The mean standard deviation is 0.2006mm, which is negligible for anatomical study of rat bones.

Validation of the tracking of the bones

To validate the marker-based tracking of the bones, we compare our method to the SVD-based tracking of [BBG+10]. Indeed, as there are 4 markers on the scapula, 4 markers on the humerus and 4 markers on the radius-ulna, we can use the rigid transformation of each set of markers to reconstruct the motion of the different bones. We can as a result test our method against a method proven to be accurate.

In our method, as the markers are implanted on the bones and therefore all rigidly connected to the bone, the stiffness of the linear springs is the same for all markers. For the same reason, the stiffness in rotation of the angular springs is chosen low. As the joints involved are assumed to be ball-and-socket joints, the stiffness in translation of the angular springs is chosen high.

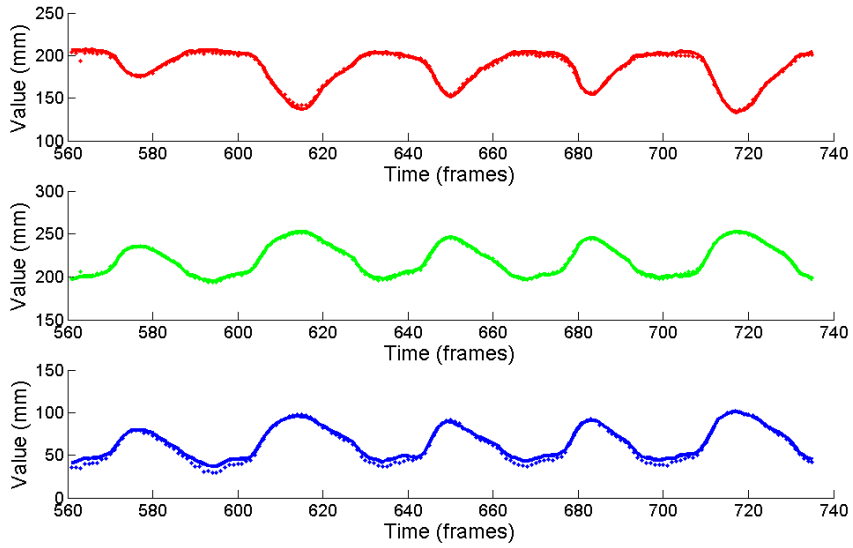
One measure of the correctness of the tracking is the elongation of the linear springs. The elongations for the 5 sequences are reported on Table 2.2. Observing these distances over time, spikes in the elongations can be seen (bottom of Figure 2.33). One way to diagnose these errors is by looking at those side by side with the 2D tracking error computed to validate the 2D tracking of the markers. We can see on Figure 2.33 that the spikes in the elongations match errors in the 2D tracking. These spikes are thus more likely to come from tracking errors than from a fault in our tracking method.

To validate our method, we also compare our results by tracking the bones using the University of Brown Xromm tools presented in [BBG+10]. Quantitatively, we can compare directly the transformations computed by the two methods. Figure 2.34 (respectively Figure 2.35, Figure 2.36) shows the position and orientation of the scapula (respectively humerus, radius-ulna) obtained with both methods for one of the 5 sequences. If the results seem similar, our solution appears to give smoother results (see the rotation about Y of the radius-ulna for example). The only major difference is the peaks for the humerus that have a smaller amplitude with our method. This can be seen for instance around frame 660 on Figure 2.35. Figure 2.37 shows the reconstruction for both methods on frame 660. One can notice that, unlike the SVD-based method, our method does not lead to penetration between the scapula and the humerus. Following [WAK+10], we assess the similarity between both results by computing the correlation coefficient between the trajectories of the degrees of freedom of the model. Table 2.3 shows those correlation coefficients for the different anatomical entities. The trajectories can be considered equivalent.

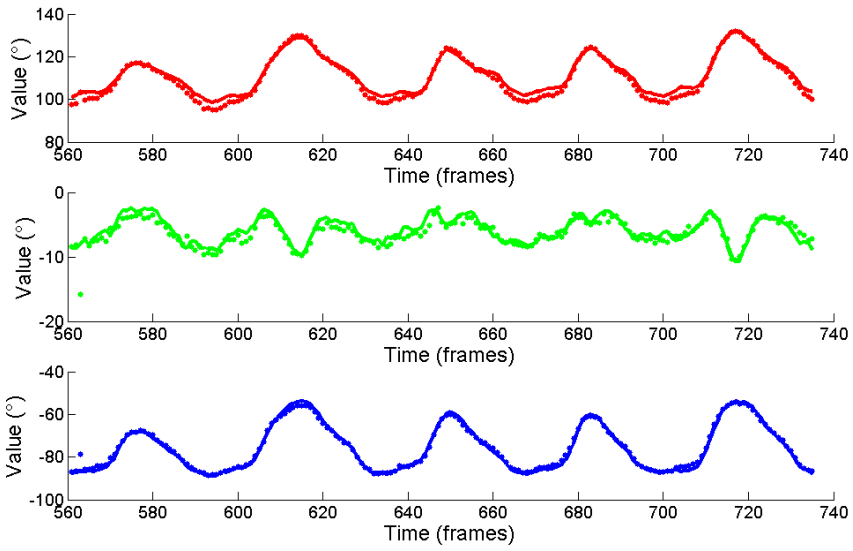
	Tx	Ty	Tz	Rx	Ry	Rz
scapula	0.9958	0.9943	0.9976	0.9958	0.9993	0.9897
humerus	0.9917	0.9495	0.9719	0.8753	0.9810	0.9761
radius-ulna	0.9829	0.9742	0.9781	0.8570	0.9789	0.7302

Table 2.3: *Correlation coefficients of the degrees of freedom of the solids recovered with rigid transformations between sets of markers and recovered with our method for* $n = 736$ *frames.*

Qualitatively, we can however point out a few differences between both methods. As explained in Section 2.4.2, for long thin bones, placing 3 non-collinear markers can be difficult and leads to ill-conditioned problem with an SVD-based model. As shown in Figure 2.25, this creates noise on the roll of the radius-ulna whereas the angular springs in our method regularises the solution and prevents that noise. Overall, our method gives less jittery results by preserving natural articulation constraints. It is less sensitive to noise coming from the tracking of the markers or from the fact that the positions of the landmarks are estimated and not extracted from the CT-scan. Qualitative comparisons of SVD-based methods and our method can be seen on Figure 2.38 and 2.39.



(a) Translation X (red), Y (green), Z (blue)

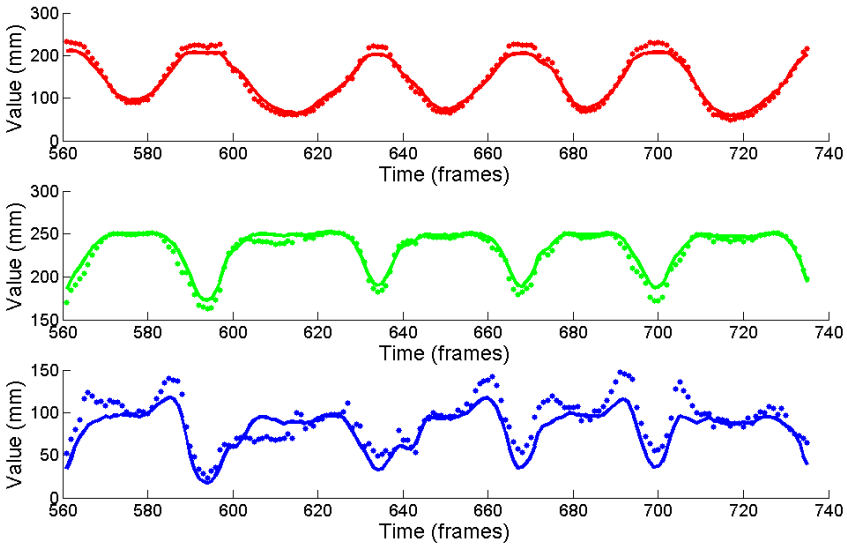


(b) Rotation X (red), Y (green), Z (blue)

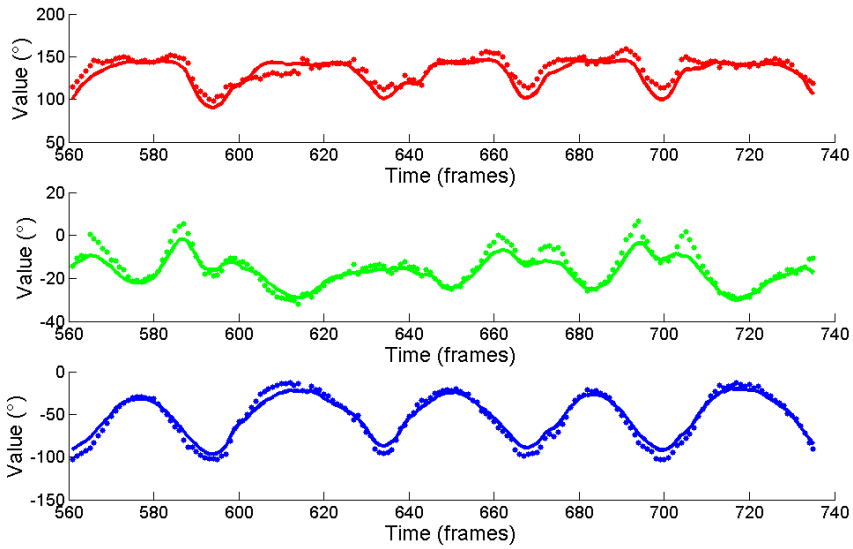
Figure 2.34: *Comparison of the tracking of the scapula with rigid transformations between sets of markers (dots) and our method (line).*

Study of the influence of the number and location of the markers

This experiment also allows us to study the influence of the number of markers as well as the influence of where the markers are implanted on the quality of our tracking. Unlike methods that rely on at least 3 markers per bone to compute the rigid transfor-



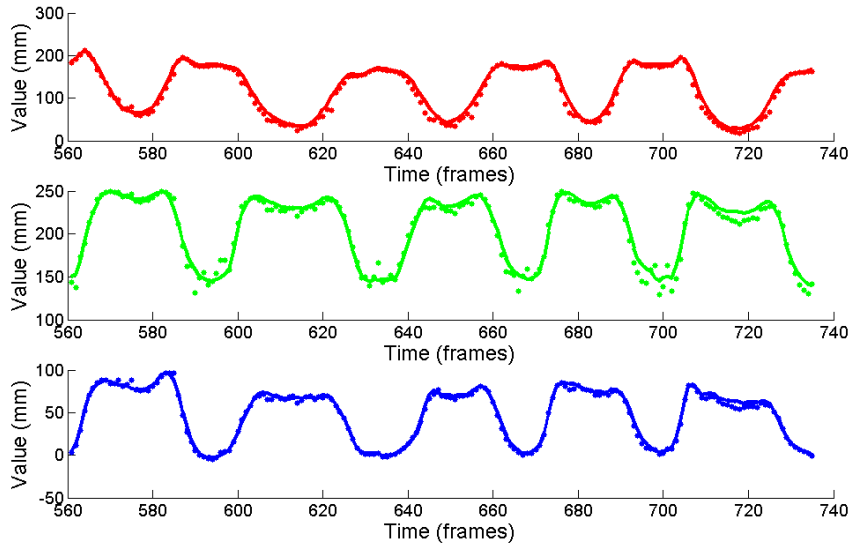
(a) Translation X (red), Y (green), Z (blue)



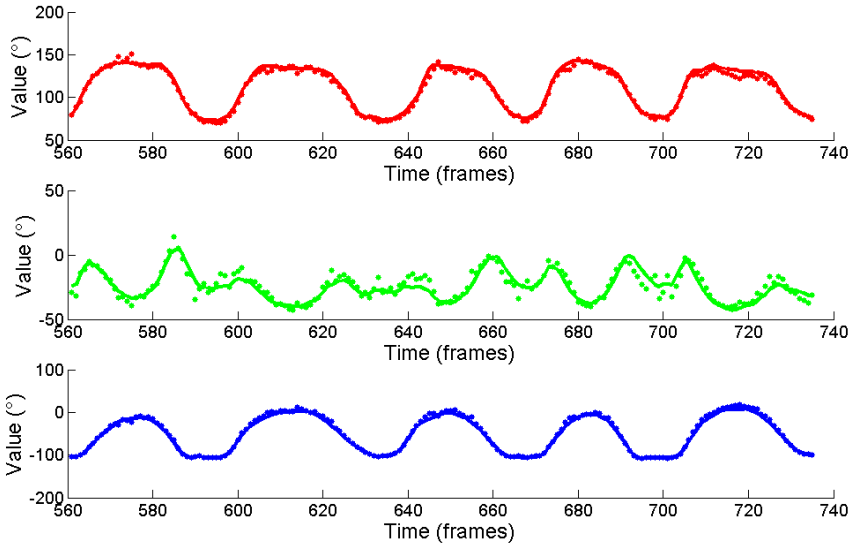
(b) Rotation X (red), Y (green), Z (blue)

Figure 2.35: *Comparison of the tracking of the humerus with rigid transformations between sets of markers (dots) and our method (line).*

mation of a bone, our method does not require a certain amount of markers. However, the number of markers will influence the precision of the tracking. We study this influence using the animation with all markers (4 per anatomical entity) presented above as ground truth.



(a) Translation X (red), Y (green), Z (blue)



(b) Rotation X (red), Y (green), Z (blue)

Figure 2.36: *Comparison of the tracking of the radius-ulna with rigid transformations between sets of markers (dots) and our method (line).*

To do so, we decrease the number of markers used to reconstruct the movement of the animal and study the difference between this tracking and the ground truth. We also measure the distance between the landmarks and the tracked 3D markers for the markers that are not used in the weighted IK. Table 2.4 sums up the results. We can see, logically, that using 3 or 4 markers does not make a significant difference. Even

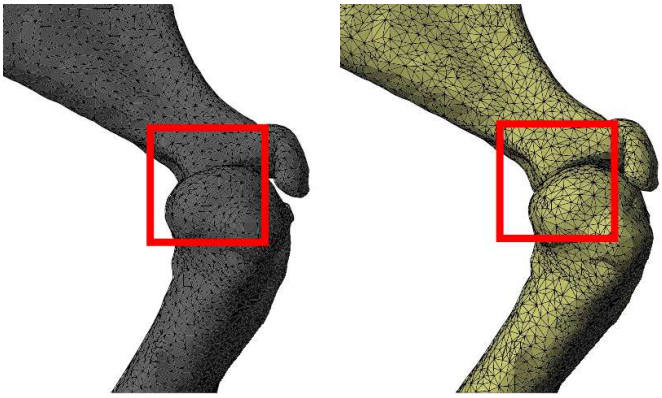
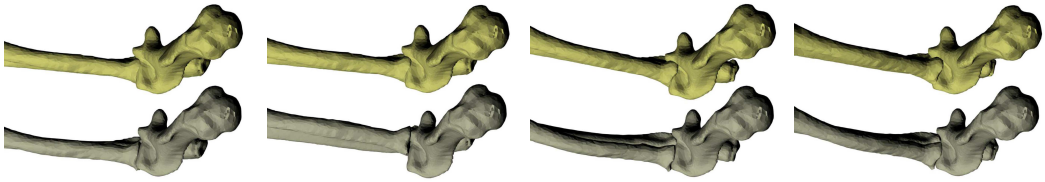
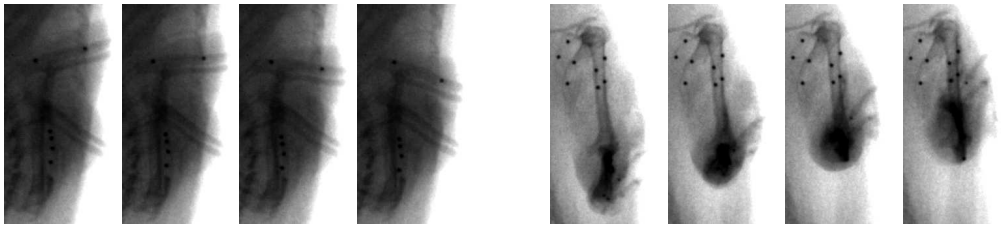


Figure 2.37: *Comparison of the scapula-humerus articulation tracked with rigid transformations between sets of markers (left) and our method (right) on frame 660.*



(a) **(b)** **(c)** **(d)**



(e) Lateral view **(f)** Ventrodorsal view

Figure 2.38: *Comparison of tracking using rigid transformations between sets of markers (grey) and our method (yellow) on 4 consecutive frames (a-d). One can notice the sudden shift of the radius that occurs between (a) and (b) as well as its sudden twist that occurs between (b) and (c) with the SVD-based method. On the other hand, our method produces a smooth natural tracking. Bottom row shows the frames used for tracking.*

with 1 marker per bone, we can obtain a reconstruction valid within 5°.

This experiment also highlights which markers lead to a more accurate tracking. We can notice on Table 2.4 that choosing marker R1 (on the elbow) or marker R3 (on the

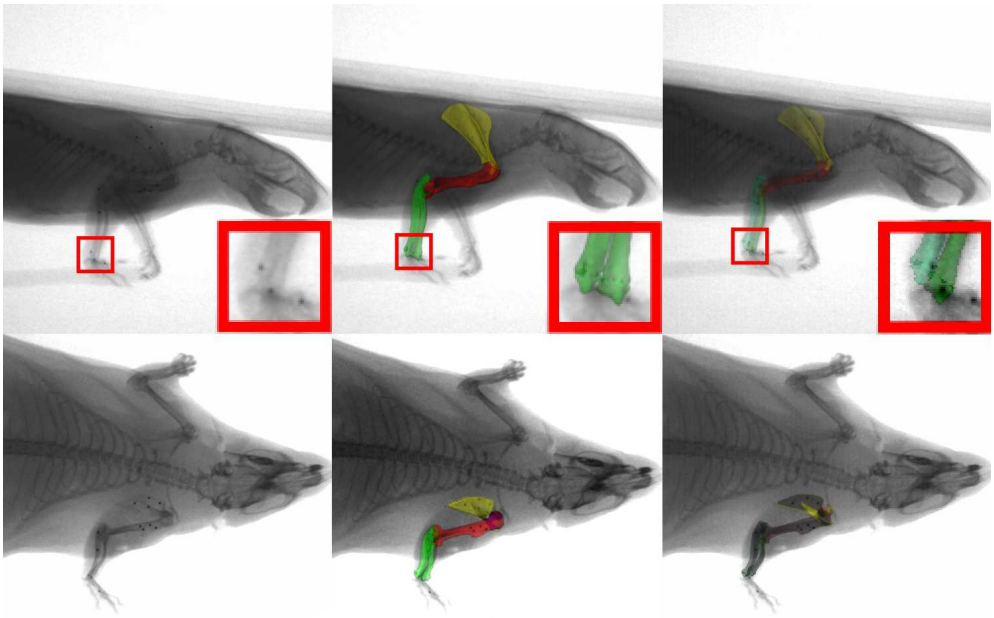


Figure 2.39: *Overlay of SVD-based tracking (middle) and our tracking (right). Notice how the radius comes out of the body at the radius/carpal articulation in the SVD-based tracking.*

radius-metacarpal articulation) greatly modifies the reconstruction (see Figure 2.29 for the name and location of the markers). The first two images in Figure 2.40 illustrate the issue raised by choosing marker R3. In the first image, the elbow is not constrained by a marker. This leads the angular springs to have a greater influence on the angle made between the humerus and the radius. Indeed, as R1 and $\text{Hi}_{i=1..4}$ are not enough to remove the ambiguity of the location of the elbow, the preferred position is one similar to the one in the rest pose. This results in a badly located elbow. This can be avoided by lowering the stiffness in rotation of the angular springs, hence lowering the influence of the rest pose (right of Figure 2.40).

This process gives us an experimental protocol on how to loosen experimental constraints by lowering the number of markers. Indeed, we have shown that we do not require as many markers as SVD-based methods to track the bones. However, not all locations for markers lead to the same accuracy. One could first do an experiment with many markers on a cadaveric animal, test which ones are important and then lead other experiments on live animals with a less invasive experimental protocol by only implanting the markers required to fall within a certain accuracy.

The robustness of our method can be used to reconstruct bones on which 3 non-collinear markers are hard to implant. On the data from the cadaver rat, there is only one marker on the metacarpal bone of the fourth digit. With our technique, we can

Markers used	Bones	$\|T_{Xromm} - T_{Sbromm}\|$ (mm), difference in Rx,Ry,Rz (°)	Max d(landmark,marker) (mm)
S1 S2 S4	scapula	0.1511, 0.4541 0.3271 0.6269	
H1 H2 H4	humerus	0.1369, 0.3932 0.3768 0.3573	1.8969
R1 R2 R3 M1	radius	0.3012, 0.6228 1.0359 0.9641	
S1 S3	scapula	0.8965, 2.1927 2.3722 3.8907	
H1 H4	humerus	0.4446, 1.1578 1.1019 1.5233	2.1768
R1 R3 M1	radius	0.2877, 0.5563 0.9345 0.9878	
S1 S4	scapula	0.7716, 1.7582 1.4798 3.3740	
H1 H4	humerus	0.2549, 0.6268 0.8091 0.9363	1.8198
R1 R3 M1	radius	0.3230, 0.6488 1.1306 1.0537	
S1	scapula	1.2050, 2.8222 3.1287 5.1481	
H1	humerus	0.4934, 1.2698 1.1907 1.9536	2.4442
R1 R3 M1	radius	0.3048, 0.6230 1.0353 1.0366	
S1	scapula	2.7641, 4.3179 5.6471 12.8113	
H1	humerus	1.7550, 5.5889 4.8697 9.9182	8.2003
R3 M1	radius	1.6557, 5.3109 5.4406 5.6671	
S1	scapula	1.1779, 2.9475 2.9435 4.9965	
H1	humerus	0.5898, 1.6811 1.6512 1.8233	3.9305
R1 M1	radius	1.0598, 1.9516 3.3752 3.6056	

Table 2.4: *Study of the influences of the markers used for tracking for $n = 736$ frames. See Figure 2.29 for markers' location.*

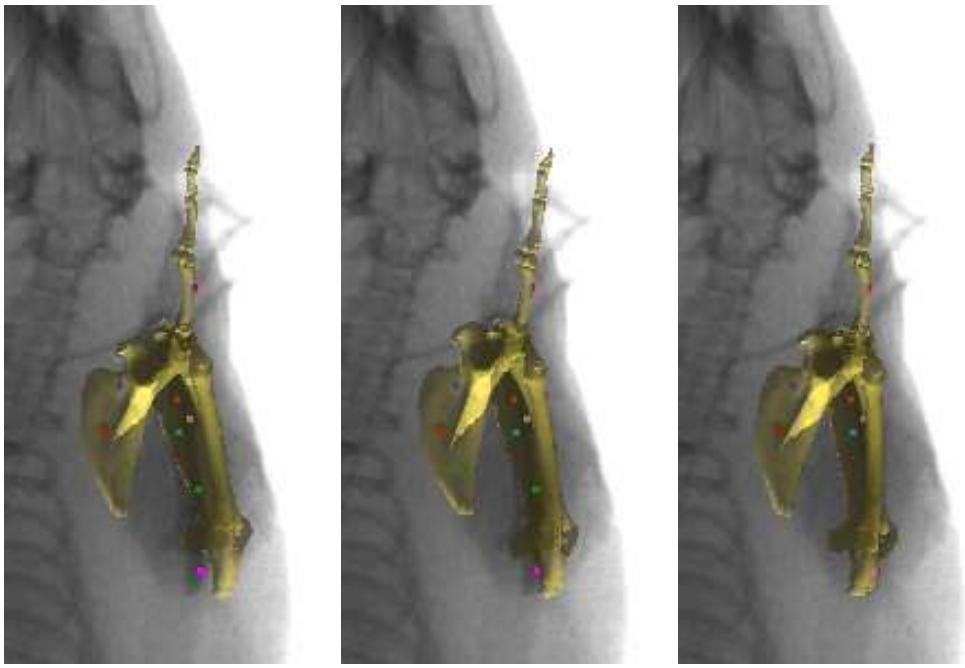


Figure 2.40: *Study of the influences of the markers used for tracking. From left to right : using marker R1, using marker R3, using marker R3 with stiffness in rotation lowered. See Figure 2.29 for markers' location.*

reconstruct the movement of this bone (Figure 2.41).

Figure 2.41: *Tracking of the metacarpal bone of the rat with only one marker on it.*

Application

Bi-planar X-ray videos has allowed us to evaluate the different steps of our marker-based tracking (Figure 2.6) : calibration, tracking of markers, influence of the number and location of landmarks, weighted IK. However, this is an ideal set-up with several X-ray videos and markers implanted on the bones, which is not always available. We therefore try to reproduce the same experiment but on our locomotion platform, presented in Section 1.2.2 by tracking the foreleg of a rat running on a treadmill.

The set-up consists of 4 cameras and 1 lateral cineradiography. As studied before [BC10][FPC+06], reconstructing motion from markers on the skin suffers from the fact that those markers are not rigidly connected to the bones. As such, SVD-based methods result in largely incorrect results. However, with our weighted Inverse Kinematics method that relies on two types of constraints (anatomical and tracking), we can obtain valid results (Figure 2.42). Figure 2.43 shows the flexion-extension at the articulations as well as the height of the wrist for 2 cycles of locomotion for both the bi-planar set-up (left) and the treadmill set-up (right). The movement is not expected to be the same in both cases as a manual simulation of locomotion is not the same as a natural cycle of locomotion. However, we can see that the movements are consistent.

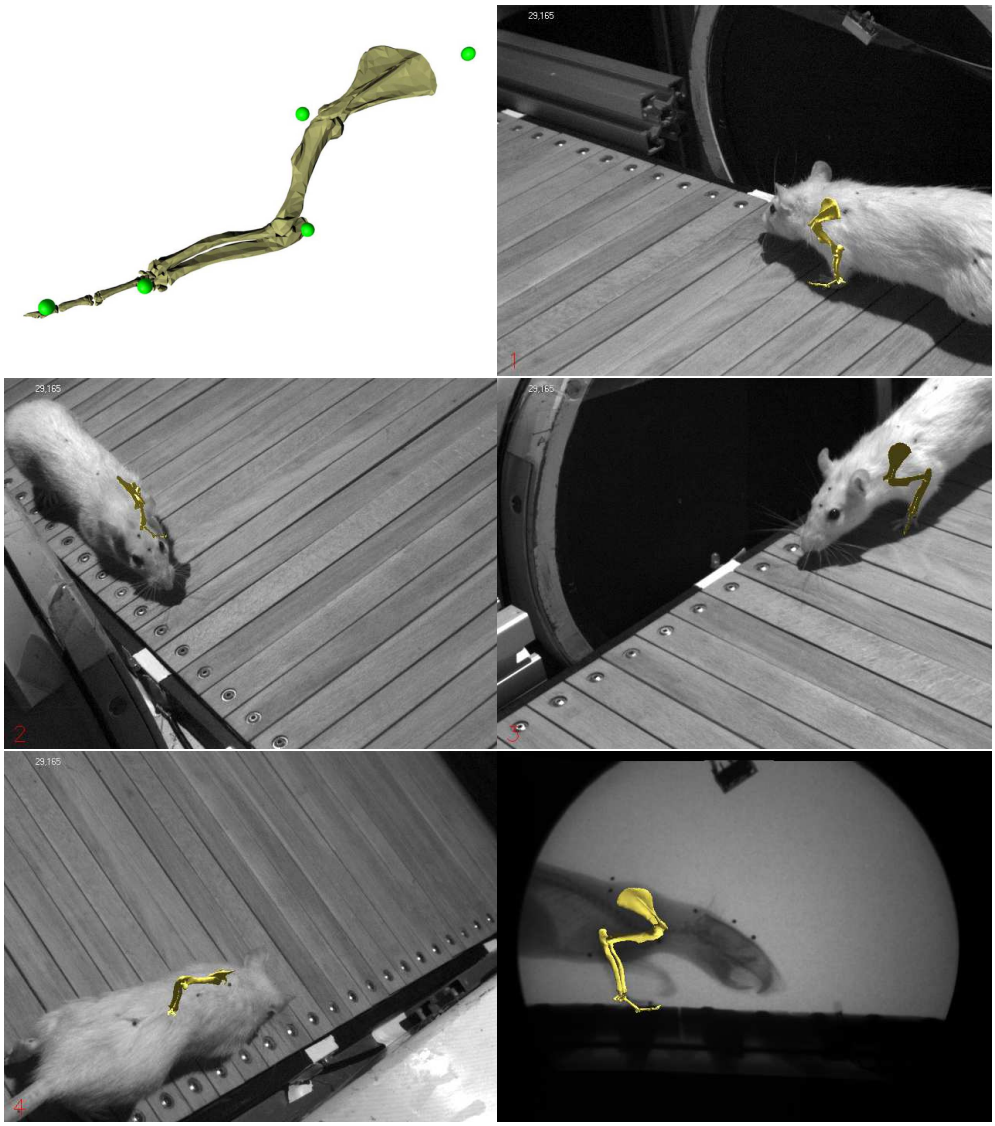


Figure 2.42: *Tracking of rat locomotion. The green spheres in the top left image are the landmarks of the bones.*

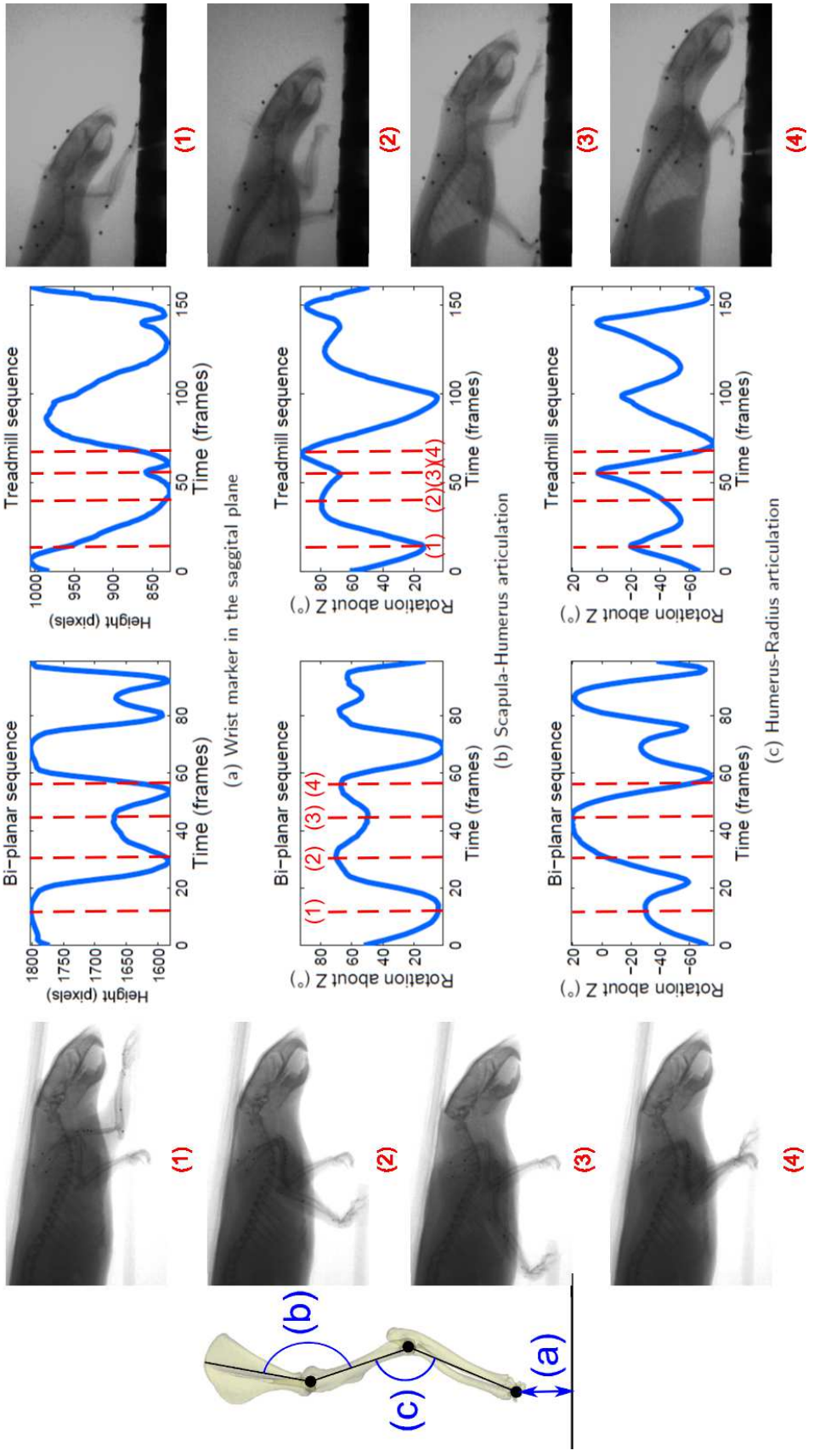


Figure 2.43: *Comparison of the articulations on 2 cycles of locomotion : one simulated on a cadaveric rat (left) recorded by bi-planar X-rays and one from a live rat on a treadmill recorded by external videos and one X-ray video (right).*

2.5.2 Results for various morphologies and motions

The process having been validated on the foreleg, we use the whole model as described in Section 2.3 on two types of motion : locomotion and swimming. Results on the locomotion sequences (ratC6X1M1lo and ratC4X1M1lo) can be seen on Figure 2.44 and Figure 2.46. For those sequences, an X-ray video is available as well as makers surgically implanted directly on the bones. Those markers can not be tracked in 3D, thus they can not be used in our tracking, but their 2D positions can be tracked on the X-ray video. There are not enough markers to be able to conduct a comparison to SVD-based methods but our tracking can be evaluated by comparing the 2D reprojections on the X-ray of landmarks on the bone and the true positions of the markers as tracked on the X-ray video. For ratC6X1M1lo, the mean error of these reprojections over 300 frames (6 cycles of locomotion) for the 9 internal markers is 15 pixels. At the depth of the rat, this is equivalent to 2.4mm. Figure 2.45 shows a comparison of the position of the simulated landmarks and the ground truth (tracked markers) for one cycle of locomotion. One can see that the markers on the humerus (left) and the tibia (right) are very close to ground truth whereas they were not used for tracking. In the middle, one can see that one of the markers of the spine is pretty far from ground truth. However, the trajectory is the same. This bias is explained by the fact that the landmark was not properly located on the animation model. As this marker is not used for tracking, it has no impact on our results. For the sequence ratC4X1M1lo, the mean error for the 17 internal markers over 375 frames (4 cycles of locomotion) is 10 pixels, which is equivalent to 1.6mm.

On the swimming sequence ratC5X0M1sw, no X-ray video is available. Figure 2.47 shows our tracking from the 15 markers. The distribution of the markers is : 4 markers on the head, 3 on the spine, 1 on each shoulder, thigh, wrist and fifth metatarsal. This leads to only 8 anatomical entities with makers on them. For comparison, in the locomotion sequences, about 15 anatomical entities had external markers attached. It can qualitatively be seen that this leads the tracking to be less precise. Therefore, even though our method is more robust to a low number of markers than previous methods to precisely and accurately track internal structures of animals, the precision of our method is still dependent on the number of markers.

In the context of this thesis, the method has been extensively used and validated on rodents. However, nothing in our method is specific to rodents. For any species, an animation model can be derived as in Section 2.3 and our tracking method used to recover the state of the animation model from videos. Figure 2.48 shows an example of tracking zebra finches during take-off and landing.

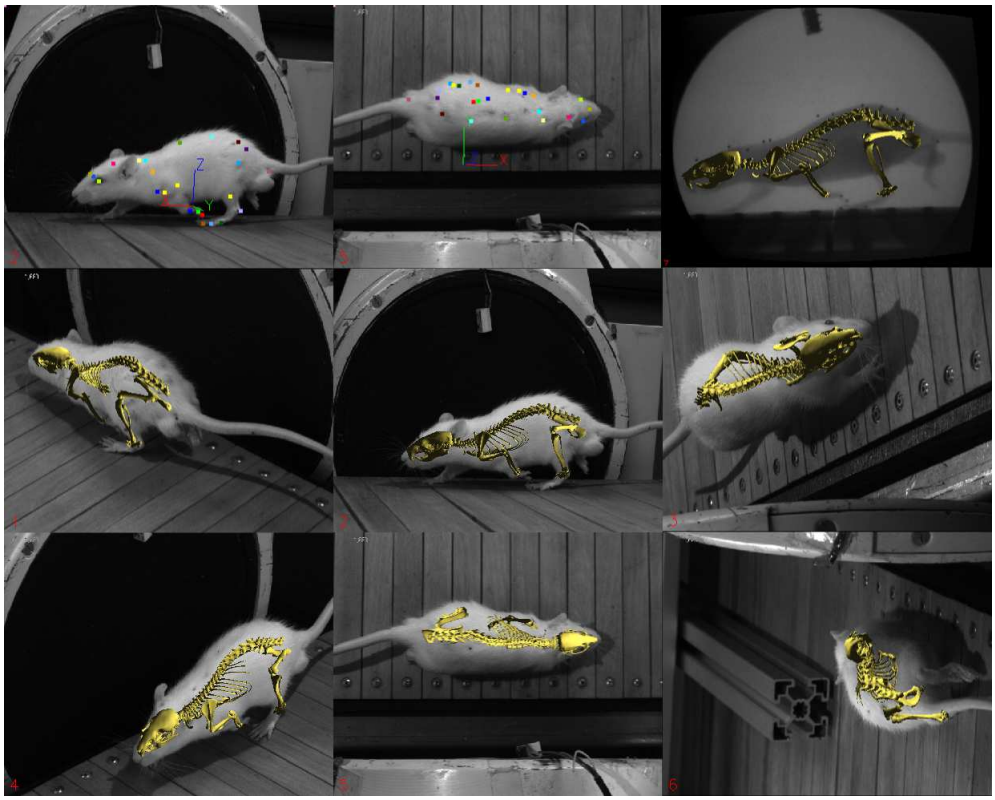


Figure 2.44: *Tracking of rat locomotion (half body).*

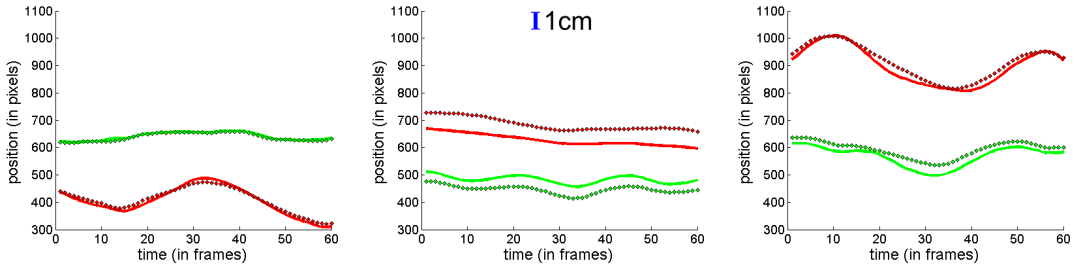


Figure 2.45: *Horizontal (red) and vertical (green) coordinates of the reprojection of the landmarks (line) on the X-ray compared to the tracking of the corresponding internal markers (dots) on the X-ray video for sequence ratC6X1M1lo. From left to right - marker 13 (humerus), 16 (spine) and 24 (knee). See Figure 2.22 for the exact location of the markers.*

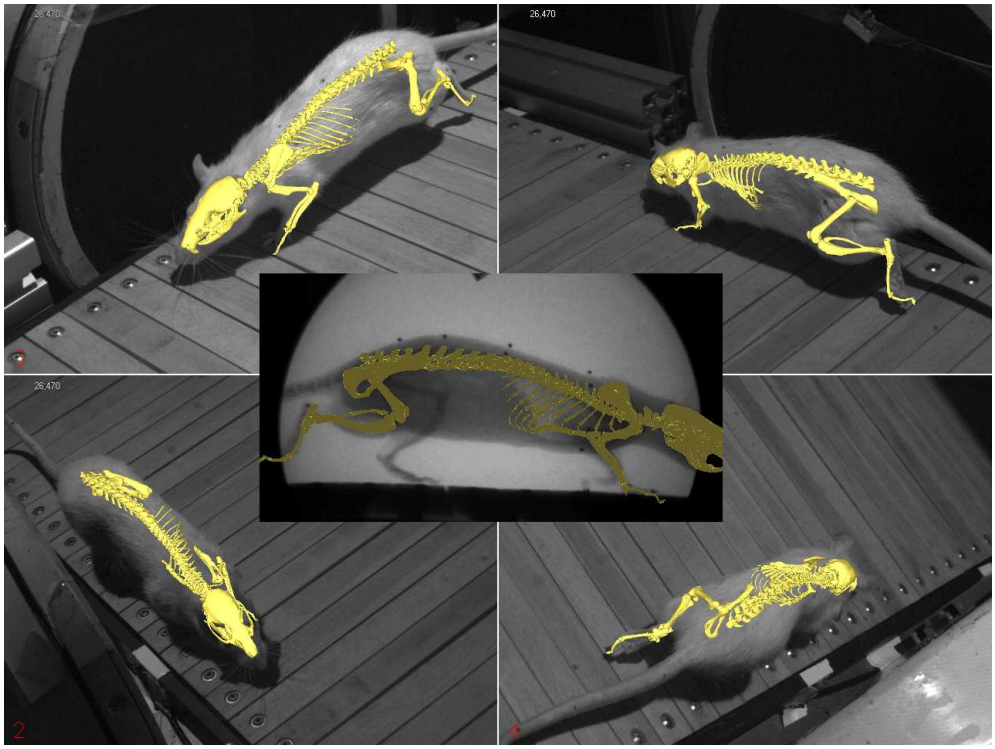


Figure 2.46: *Tracking of rat locomotion (half body).*

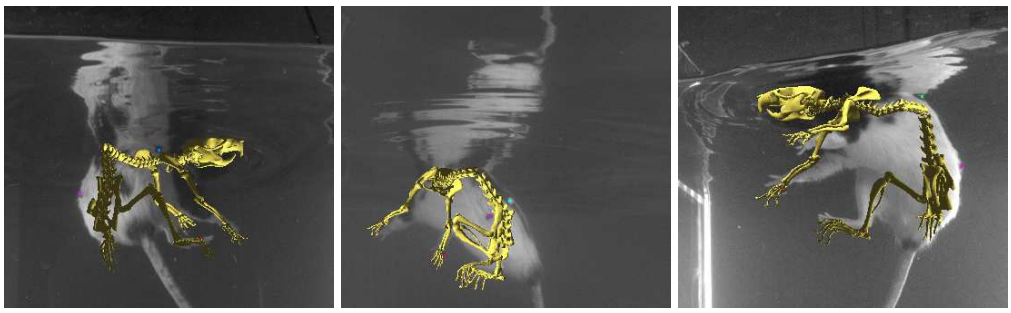


Figure 2.47: *Tracking of rat swimming.*

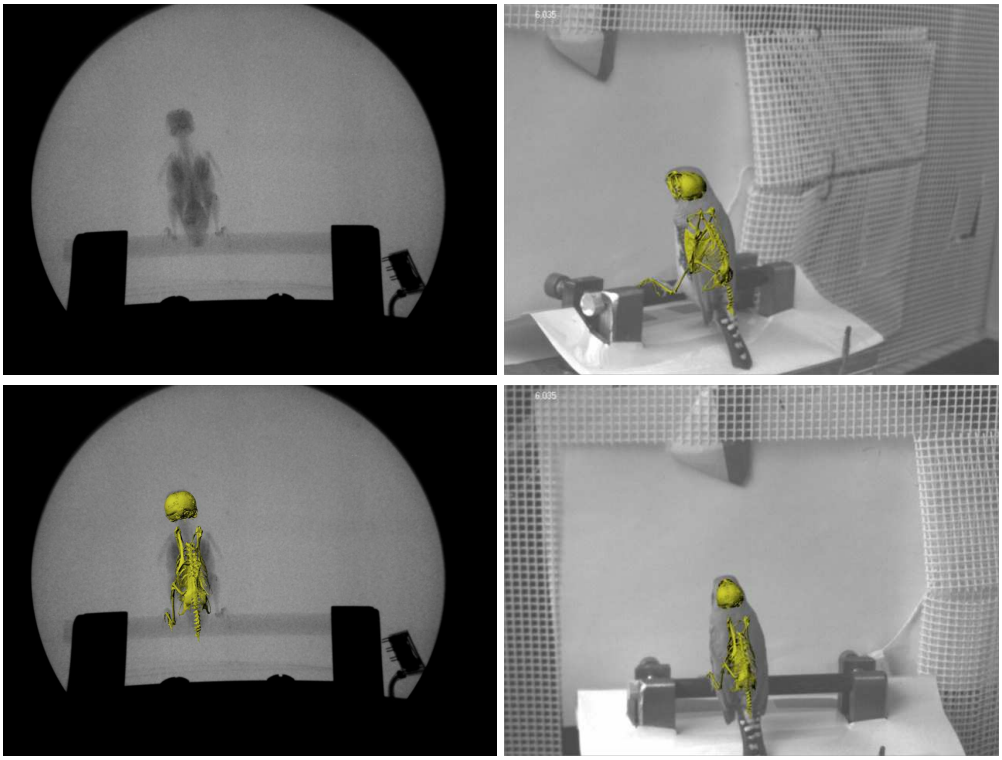


Figure 2.48: *Tracking of zebra finch landing.*

2.6 CONCLUSION

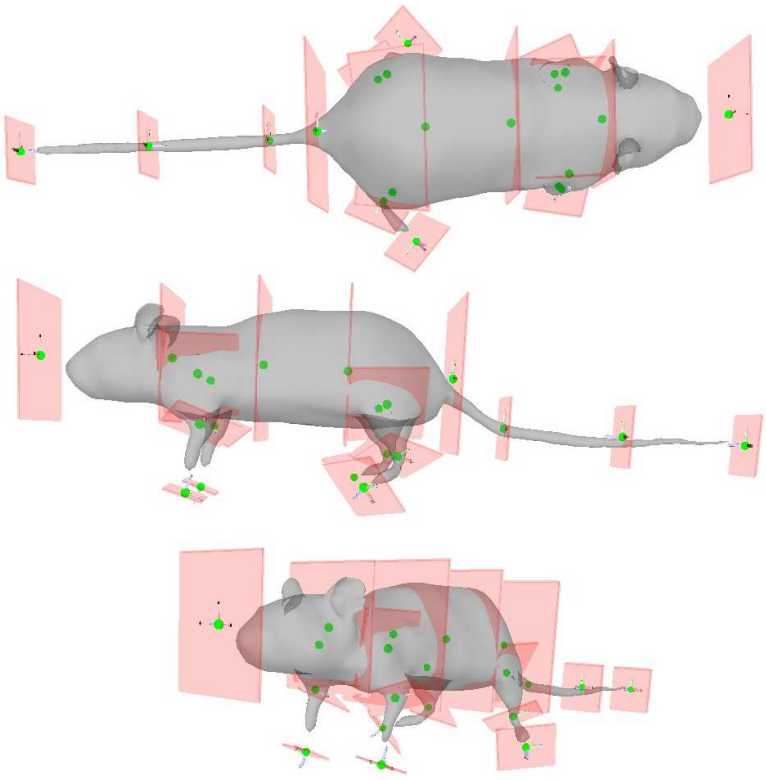
We have presented a model for the skeletal structures of rodents that can be animated with weighted Inverse Kinematics using markers on the skin. Thanks to the weighting system and the use of anatomical constraints, we have been able to overcome the artefacts usually seen when using markers on the envelope of the animal. This has been validated with the use of X-ray videos. However, a perfect validation would have been to use a set-up with both standard cameras and external markers (as in ratC6X1M1lo) and bi-planar X-ray videos and internal markers (as in ratC0X2M1lo). Unfortunately, such a set-up was not available to us.

Still, we have demonstrated that our method is reliable and robust to skin sliding. Nevertheless, markers are not always available or desirable. Most often than not, the only cue available is images of the envelope. But given what we have demonstrated in this chapter, if we are able to track the skin from those images with a temporally coherent mesh, we can track the internal structures by using points of this mesh to simulate markers. To do so, we therefore need a temporally consistent envelope that deforms over time.

CHAPTER

3

ACQUISITION AND DEFORMATION MODEL OF THE ENVELOPE.



In the previous chapter, a motion model of the internal structures of rodents was derived. As explained, this model can be registered on videos using marker-based approaches. However, in many contexts, the experimental conditions do not allow for markers to be used. For instance, in large-scale experiments when many animals are to be tested, it becomes tedious to implant markers on every animal. Moreover, the animals are usually successful in removing markers, which means that the recording has to take place immediately after the markers are implanted and can not be re-tried later. All this points to the fact that marker-based approaches are limited to punctual studies on a few animals but can not be applied to large-scale behaviour monitoring.

This leads us to consider marker-less reconstruction. This means studying the internal structures through observations of the envelope in a multi-view set-up. In our case, what we see and therefore can observe on images is the fur, which means that the envelope is the surface created by the fur of the animal. As can be seen on Figure 3.1, this surface is an irregular surface that is linked to the bones through anisotropic hairs, skin and fat. Mapping observations of the envelope to internal structures is therefore difficult.

In this chapter, we will first review computer vision techniques to recover the envelope from multi-view set-ups as well as deformation models that can be used to model this envelope. Then, we will describe the specific data-set that we are studying. Based on this data-set, we will build an appropriate deformation model. We will finally see how to link this external model to the internal model described in Chapter 2.

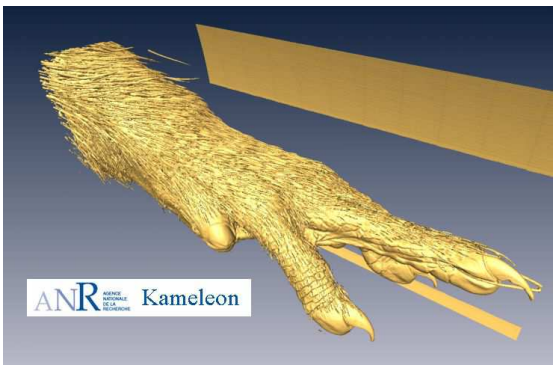


Figure 3.1: *CT-scan of a rat at resolution 50μm.*

3.1 PREVIOUS WORK

3.1.1 Multi-view reconstruction

In order to recover the geometry of the envelope, we use a marker-less multi-view set-up as a capture device. Extracting dense information from several cameras is being studied extensively in computer vision. The different techniques can be classified depending on the kind of information they extract from the images : depth maps or geometry.

Historically, monocular vision systems have evolved to several cameras with the development of stereo-vision techniques. Indeed, from two calibrated stereo-cameras, a disparity map can be computed. Each pixel of the disparity map stores the difference in location of corresponding features in the two images. As illustrated on Figure 3.2, the closer the point is to the camera, the larger the disparity is. From this disparity map, a depth information can thus be extracted and stored in a depth map. An in-depth survey of stereo-vision techniques can be found in [SS02].

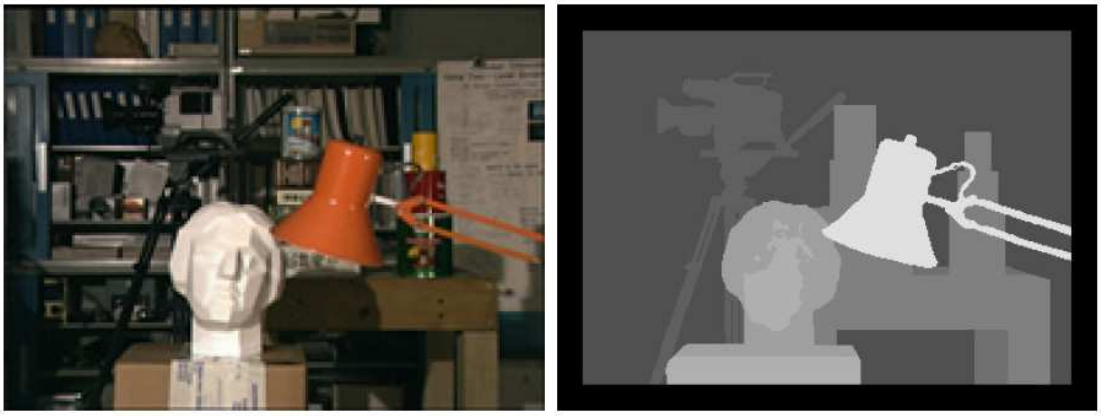


Figure 3.2: *Disparity map - left : original image - right : disparity map from [SS02].*

This however requires to find corresponding features in both images. To overcome the difficulty that stereo-vision has to tackle scenes that lack textures, structured light can be used. In structured light set-ups, patterns are projected onto the scene to create artificial textures that make the matching process easier and more robust. An example from [ZSCS04] can be seen on Figure 3.3. However, such set-ups are cumbersome and unpleasant for live subjects, even frightening for animals because the projectors tend to dazzle the subjects.

In recent years, time-of-flight cameras have been developed. These cameras directly

Figure 3.3: *An example of structured light reconstruction from [ZSCS04].*

record the depth map of the observed scene. An evaluation of the use of those cameras can be found in [KBKL09].

However, stereo-vision provides us with limited knowledge of the 3D scene as the thickness of the observed object is unknown and the surfaces occluded by the object on the foreground can not be reconstructed. For instance, the back of the head can not reconstructed in Figure 3.3.

The second kind of information that can be extracted from multi-view systems is geometry. The techniques that extract such information fall into different categories depending on the information they use : matched features or silhouettes.

In the first category, the techniques are the same as the ones used in stereo-vision. Features are matched in-between different cameras. If one camera gives us a 3D line on which the 3D point lies, the line that goes from the center of camera through the pixel on the image plane (as seen on Figure 3.4), two cameras provide two lines that can be intersected at the position of the 3D point. This process is called triangulation. In [SCD+06], a review and an evaluation of those techniques are conducted. As in stereo-vision techniques, this approach suffers from lack of textures in the scene and structured light can be used.

Another clue that can be used to extract 3D geometry from multi-view systems is the silhouette of the studied object on the images. As explained in [Lau94], the silhouettes define a 3D space in-which the object can lie. Laurentini defines the visual hull as the maximal 3D envelope of the studied object that projects back inside the silhouettes of all cameras. As a result, the studied object is completely inside the visual hull. An example of a visual hull can be seen on Figure 3.4. The visual hull is not the object itself but an approximation of it. For instance, concavities can not be recovered from silhouettes. The quality of the visual hull depends on the quality of the silhouettes and on the number of viewpoints.

One of the most common methods to compute the visual hull is to voxelise the scene

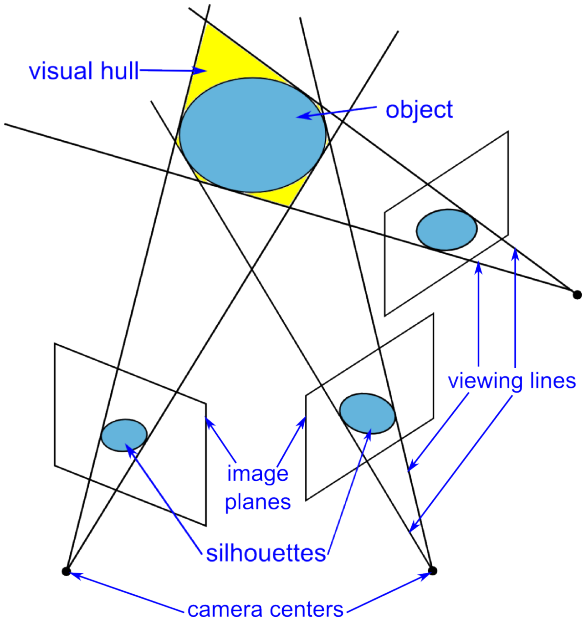


Figure 3.4: *An example of a visual hull (slice representation).*

and decide for each voxel if they belong to the object or not. A voxel belongs to the object if and only if it projects into a silhouette in all images. A review of voxel-based techniques can be found in [SMCS]. The result is a set of voxels labeled as being part of the object (Figure 3.5). The precision of the reconstruction thus depends on the resolution of the voxel grid. Most of the time, Marching Cube [LC87] is then used to build a mesh from the voxel grid in order to apply many computer graphics techniques such as texturing or smoothing to the result.

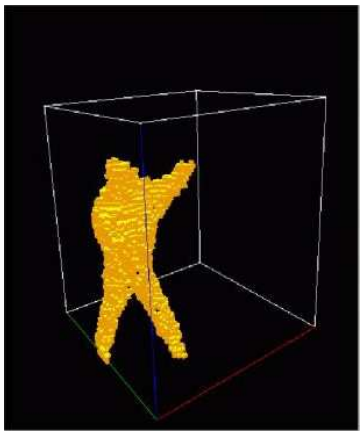


Figure 3.5: *An example of a voxel model inside the scene volume as reconstructed in [CKBH00].*

Another method relies not on the inside of the silhouettes but on its contours to directly build a surface. Indeed, as can be seen in Figure 3.4, each pixel on the contour of the silhouettes generates a viewing line going through the center of the camera and the pixel. The set of viewing lines generated from one image generates a viewing cone whose peak is the center of the camera. The intersection of the viewing cones generated from the different viewpoints is the visual hull. Unlike in voxel techniques, this visual hull is a polyhedron.

All these methods generate temporally independent reconstructions of the studied object that can be used as a template on which a deformation model can be built. The choice of this deformation model is paramount to realistic and smooth deformations.

3.1.2 Deformation model

Traditionally, in computer animation, characters (humans, animals or creatures) are animated with skeleton animation. Skeleton animation requires a mesh and a hierarchy of joints called skeleton (see Figure 3.6). Each joint (except the root of the hierarchy) is traditionally modeled as an ideal revolute or ball-and-socket mechanical joint defined by its rotation relative to its parent. As a result, it has either 1, 2 or 3 degrees of freedom and its distance to its parent is constant. The root has 6 degrees of freedom which determines the global position and orientation of the skeleton.

The mesh, called skin, is first binded to the skeleton. This operation is called skinning. To do that, the skeleton is fitted to the mesh in a rest pose. For each vertex of the mesh, a weight is set to each joint. We therefore have a weight $w_{i,j}$ for each pair of vertex and joint. This weight represents the influence of the bone j on vertex p_i : the larger the weight, the more influenced the vertex is. It can be computed automatically as a function of the distance between the vertex and the joint or attributed manually. In the end, in linear blend skinning, each vertex p_i can be expressed as the weighted linear interpolation of the transformation of the joints :

$$p_i = \sum_j w_{i,j} M_j M_{0,j}^{-1} p_{0,i} \tag{3.1}$$

where M_j is the current global transformation at joint j, $M_{0,j}$ is the global transformation of joint j in the rest pose and $p_{0,i}$ is the position of vertex i in the rest pose. It can be noticed that skinning does not deform the mesh when the skeleton is in its rest pose if and only if for each vertex i, the sum of all the weights $w_{i,j}$ is equal to 1.

Then, to animate the mesh, the animation skeleton is animated i.e. the transformations of the joints are modified, leading to a deformation of the initial mesh as a result of equation (3.1) (see bottom of Figure 3.6). Linear blend skinning is thus a simple and intuitive way to animate a mesh as a simplification of the real skin-bones complex.

Figure 3.6: *Mesh skinned on an animation skeleton : (top) rest pose - (bottom) animation of the left elbow joint and right knee joint.*

However, linear blend skinning leads to many artefacts such as collapsing joints or twisting elbows (left of Figure 3.7). This is due to the fact Equation (3.1) can be rewritten as $p_i = Qp_{0,i}$ where $Q = \sum_j w_{i,j} M_j M_{0,j}^{-1}$, which is not necessarily a rigid transformation.

Several techniques have been proposed over the years to correct these artefacts. In 2000, [LCF00] used examples to improve the deformed mesh obtained after linear blend skinning. The modifications made on the skinned mesh are stored as displacements : $\delta p_i^{ex} = p_i - p_i^{LBS}$ where p_i^{LBS} is the position obtained by linear blend skinning and p_i is the position as set in the example. Then, for any pose of the skeleton, instead of simply computing the vertex as the result of skinning, it is computed as $p_i = p_i^{LBS} + \delta p_i$ where δp_i is interpolated from all the δp_i^{ex} with respect to the pose of the skeleton. In 2003, [MG03] built on this idea but instead of storing the displacements, they learn from the examples in order to add joints to have more precise

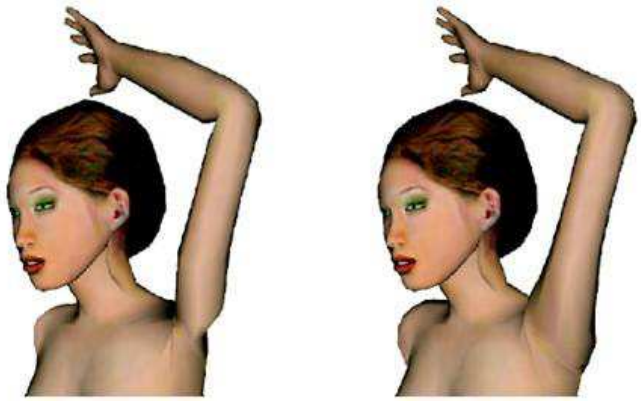


Figure 3.7: *Skinning of a virtual character using linear blend skinning (left) and using dual quaternions (right). Notice how the deformation of the shoulder is better handled by dual quaternions [KCZO07].*

deformations. For instance, to prevent elbow twisting, joints are added on the forearm. Then, classic linear blend skinning is applied on the new skeleton. However, these techniques require the intervention of an artist to create correct examples of deformation.

In [KCZO07], the heart of the problem, which is the non-rigidity of the transformation applied to $p_{0,i}$, is tackled. To do so, dual quaternions are used to interpolate the rigid transformations $M_j M_{0,j}^{-1}$ such that the result is a rigid transformation. A dual quaternion is composed of 2 quaternions :

$$\hat{q} = q_0 + \epsilon q_\epsilon \tag{3.2}$$

with $\epsilon^2 = 0$. A rigid transformation of translation (t_0, t_1, t_2) and rotation represented by a quaternion q can be represented by the dual quaternion :

$$\hat{q} = q + \frac{\epsilon}{2}(t_0 i + t_1 j + t_2 k)q \tag{3.3}$$

For a given vertex i, the weighted interpolation of the rigid transformations $M_j M_{0,j}^{-1}$ is computed as :

$$DLB(w_{i,0}, w_{i,1} \dots w_{i,n}; \hat{q}_0, \hat{q}_1 \dots \hat{q}_n) = \frac{w_{i,0}\hat{q}_0 + \dots + w_{i,n}\hat{q}_n}{\|w_{i,0}\hat{q}_0 + \dots + w_{i,n}\hat{q}_n\|} \tag{3.4}$$

where $\hat{q}_j$ represents $M_j M_{0,j}^{-1}$. This is proven to be a rigid transformation and therefore gets rid of most of the artifacts linear blend skinning generates (see right of Figure 3.7). It is thus usually used to animate humans.

Still, in the case of rodents, skinning gives unsatisfactory results, particularly on the hind legs. This is due to the fact that mice have a lot more fat that humans do. This creates an important skin sliding effect (Figure 3.8). It is therefore difficult to bind the skin to the skeleton since a vertex of the hind leg may for instance slide from in-between the knee and ankle to in-between the head of the femur and the knee. Defining the influence of a joint on a vertex by a simple weighting system as is done in skinning is not appropriate anymore.

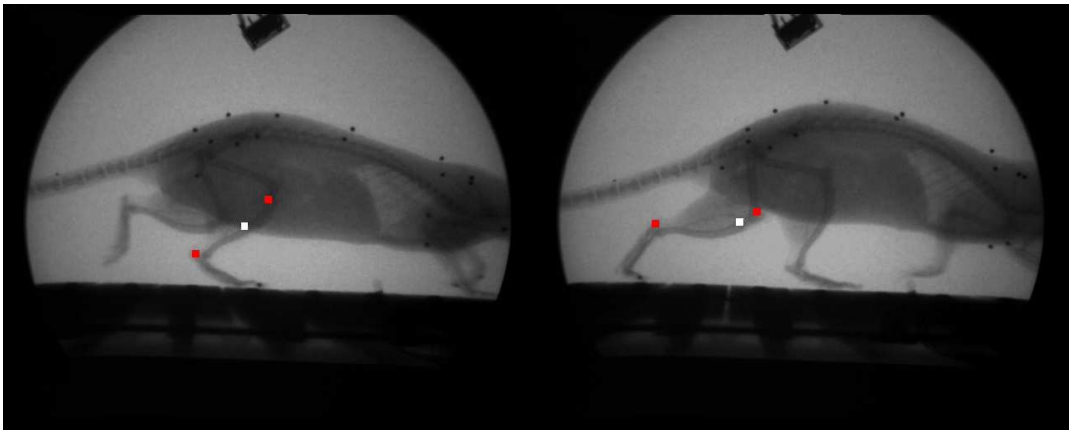


Figure 3.8: *Skin sliding effect : the markers in red are surgically implanted directly on the bones (at the ankle and at the knee) whereas the white marker is glued on the skin of the tibia.*

In recent years, cage-based animation has emerged as an extension of lattices and as an alternative to skinning. In cage-based animation, the mesh to deform, called the model, is embedded in a low-resolution mesh, called the cage. This cage controls the deformation of the model. The degrees of freedom of the deformation are the positions of the vertices of the cage $v_j, j \in [0..m]$. The vertices of the model $p_i, i \in [0..n]$, $n >> m$ are computed as the weighted linear combination of the vertices of the cage :

$$p_i = \sum_{j=0}^{m} w_{i,j} v_j \tag{3.5}$$

The weights $w_{i,j}$ are called coordinates as a reference to barycentric coordinates. The core of this method is to compute theses coordinates such that the deformation of the cage leads to an intuitive and realistic deformation of the model. In [FKR05] and [JSW05], a larger study on how to create a function over a mesh that interpolates known values at vertices, mean-value coordinates are derived and an analytical solution for piece-wise linear forms is expressed.

However, the mean-value coordinates are based on euclidean distance and do not respect such properties as interior locality and non-negativity. Interior locality means that the coordinates should decrease as a function of the distance between cage vertices and model vertices, the distance being measured within the cage (see top of Figure 3.9). Non-negativity ensures that no coordinate is negative. If a coordinate is negative then the model vertex and the cage vertex would move in opposite directions, which is counter-intuitive (see bottom left of Figure 3.9). To ensure those two properties, [JMD+07] uses the Laplace equation to derive harmonic coordinates that are non-negative and have interior locality. However, Harmonic coordinates do not have an analytical solution. A comparison between mean-value coordinates and harmonic coordinates can be seen on Figure 3.9.

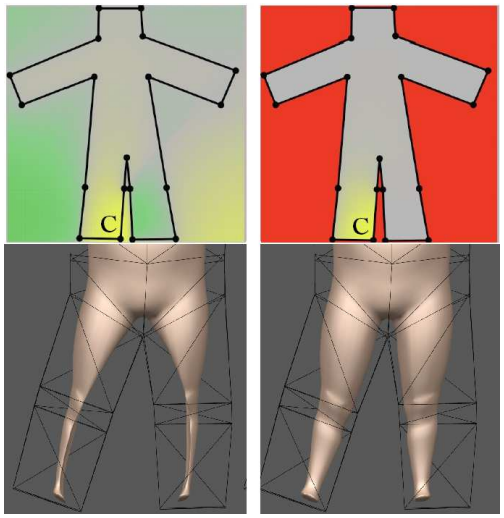


Figure 3.9: *Comparison between mean-value coordinates (left) and harmonic coordinates (right) from [JMD+07] - top row : coordinate value for vertex C, yellow indicates positive values, green indicates negative values, grey indicates null values and red indicates undefined values - bottom row : example on a 3D model, note how moving the left leg moves the right leg in the opposite direction when using mean-value coordinates.*

In [LLCO08a], Green coordinates are derived from the third integral identity of Green, which is derived from harmonic functions. Green coordinates are proven to be conformal in 2D, as well as quasi-conformal in 3D. This ensures that angles are preserved and therefore leads to nice properties in terms of shape preservation and smoothness (Figure 3.10).

Cage-based animation is thus more adequate to deform the envelope of rodents. This

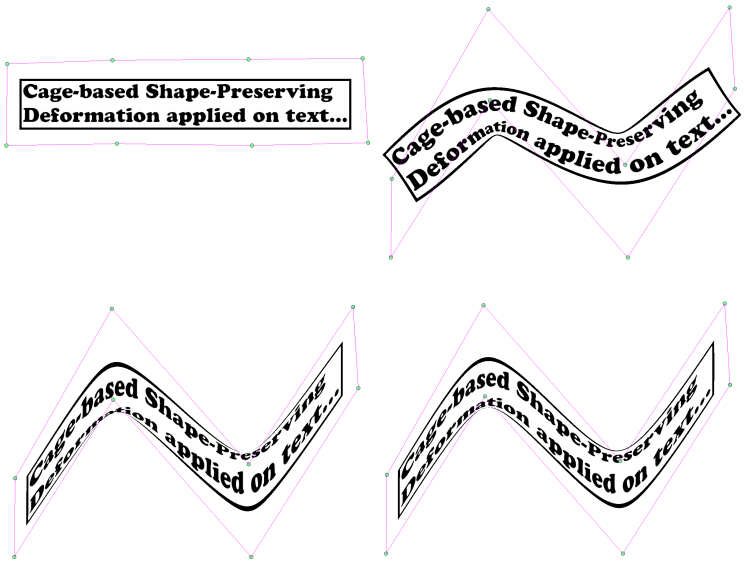


Figure 3.10: *Deformation of the text top left using Green coordinates (top right), mean-value coordinates (bottom left) and harmonic coordinates (bottom right). Notice how the text is better preserved using Green coordinates [LLCO08b].*

envelope is what can be observed from external cameras, which is the surface of the fur of the animal. This surface can not easily be modeled, we first have to extract it directly from videos. As shown in Section 3.1.1, the choice of a multi-view reconstruction technique is highly dependent on the experimental conditions : both on the set-up and on the subject. As a result, we will now focus on a specific data-set.

3.2 ACQUISITION OF THE ENVELOPE FROM VIDEOS

Multi-view reconstruction is a vast field of research and many methods have been developed over the years to extract shapes from videos. Most methods rely heavily on the scene to reconstruct and on the multi-view set-up as explained in Section 3.1.1. Consequently, we will now focus on a specific dataset from which we aim at extracting the deformation of the envelope of rodents. We focus on the set-up designed to study vestibular control of mice in micro-gravity, sequences moC8X0M0pf, to explore how state of the art techniques in multi-view reconstruction perform on our dataset.

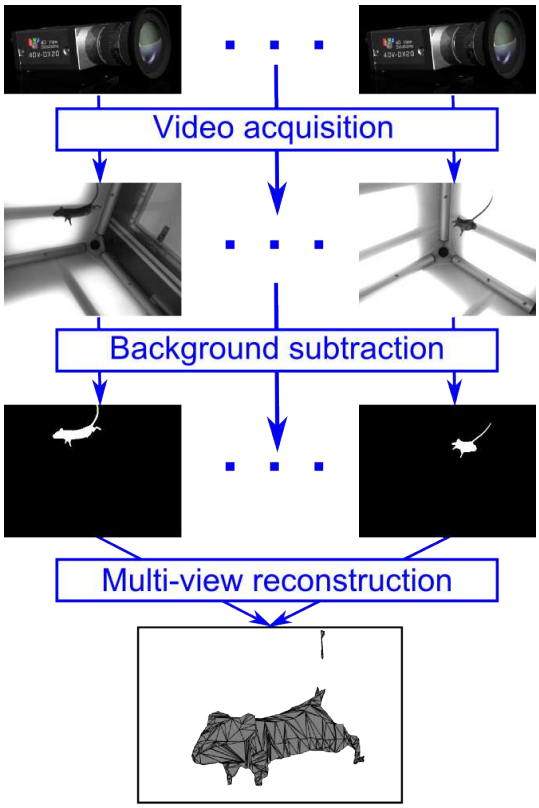


Figure 3.11: *Video processing pipeline.*

From the specific dataset obtained during the parabolic flights, we wish to extract 3D geometrical information about the envelope of the mice. As explained in Section 3.1.1, we can either match features or use silhouette informations. As the fur of mice lacks texture and structured light can not be used in our set-up, we wish to use silhouette information. Figure 3.11 illustrates the data processing pipeline to extract and use these silhouettes. After having synchronised and calibrated all the cameras so that they are

defined in a common space and time, the images are segmented to obtain the silhouette of the mouse, which is then used for multi-view reconstruction.

3.2.1 Background subtraction

The parabolic flight set-up was designed to facilitate background subtraction. Indeed, lighting and materials were selected to avoid shadows and reflections as well as moving background due to activity around the set-up. Figure 3.12 shows the evolution of the set-up to improve background subtraction leading to DEL panels covered by anti-reflexive material on the faces of the free-floating cube.

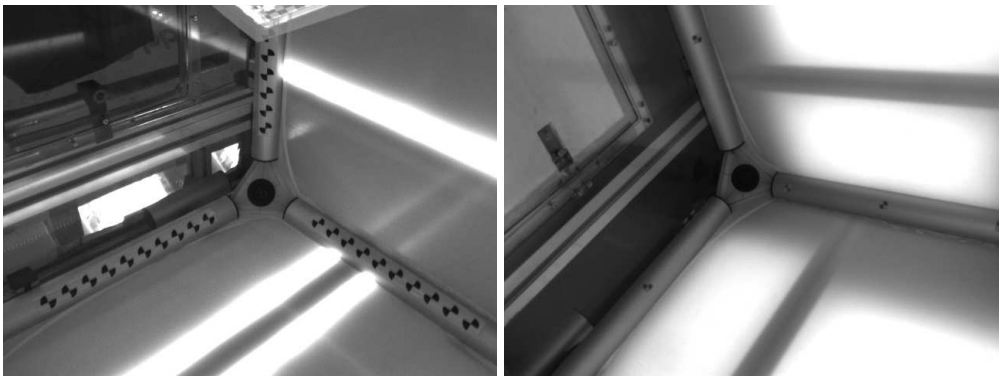


Figure 3.12: *Evolution of the set-up to improve background subtraction : from a set-up with reflections and view of the outside (left) to a better controlled environment (right).*

Several options have been tested to define the background in our dataset. Once this background is defined, for each pixel of the image, we want to define if it belongs to the background or to the foreground i.e. the mouse. This leads to computing a binary image where pixels with value 1 are foreground and pixels with value 0 are background. The user can define one or several frames as background. If several are defined, single background subtraction is done with all frames and a criterion to reduce false positives is chosen : a pixel is in the foreground if it is foreground against all backgrounds. Indeed, false positives are more likely to be corrected than false negatives as false negatives usually happen when the mouse is in front of dark parts of the background such as the objectives of the cameras and using several backgrounds will not solve the problem. Avoiding false positives on the other hand reduces noise that occurs especially on the top panel of the free-floating space where the trapdoor is and where the outside environment can be seen (see top left of the images in Figure 3.12). When images are chosen by the user for background subtraction, a pixel belongs to the foreground if its grey-level in the current image differs more than a certain threshold from its value in the background image. The threshold is constant for the whole image. One can also learn a mean background and its variance throughout a sequence of frames. In this case, a pixel belongs to the foreground if its grey-level in the current

image differs more than a certain threshold from its value in the mean image. This time, the threshold is computed automatically on each pixel as a certain times the variance at this pixel. Experiments show that both cases give equivalent results.

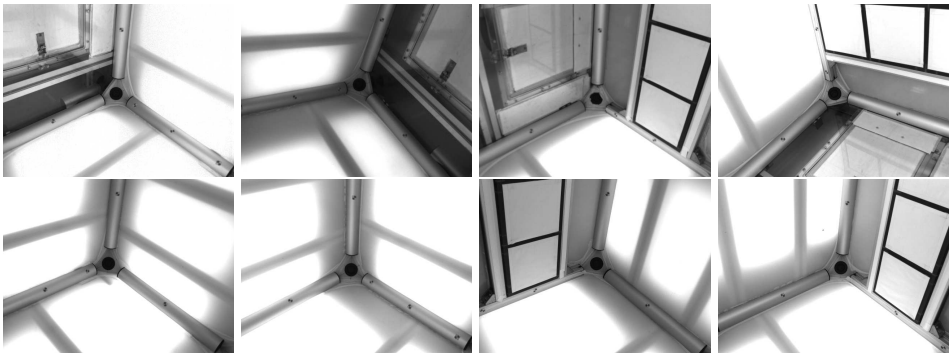


Figure 3.13: *Background used for background subtraction for the 8 cameras of the set-up.*

Figure 3.13 shows an example of user-defined backgrounds. Figure 3.14 shows examples of background subtraction with various satisfactory results. One can see that most cases of failure come from either the dark objective of the cameras preventing the dark mouse from being detected as foreground or from the uncontrolled environment on the top panel of the free-floating space that can not be covered in the same material as the other panels due to the necessity to see the free-floating space to get or put the mouse in its cage in-between series of parabola. However, segmentation of grey-level images with challenging backgrounds is out of the scope of this thesis. Yet, Section 4 will focus on tracking methods that are robust to such noise.

3.2.2 Multi-view reconstruction

Once the silhouettes are extracted, we have to choose between a volume or a surface as a representation of the reconstructed model as explained in Section 3.1.1. Whereas both techniques show the same quality of results (see Figure 3.15), we choose to model the envelope as a surface as a majority of methods in animation and shape analysis are based on surfaces. We use the Exact Polyhedral Visual Hull method (EPVH) [FB03] to obtain the visual hull polyhedron. The advantage of the EPVH method is that it is exact, that is to say that the re-projection of the reconstructed model on the images superimpose exactly with the silhouettes.

In [FB03], to compute the visual hull, a coarse approximation is first computed by only extracting a sub-set of the polyhedron edges : the viewing edges. The viewing edges are the intervals of a viewing line (Figure 3.4) that re-project in silhouettes in all other images (see Figure 3.16). From these viewing edges, the other vertices and

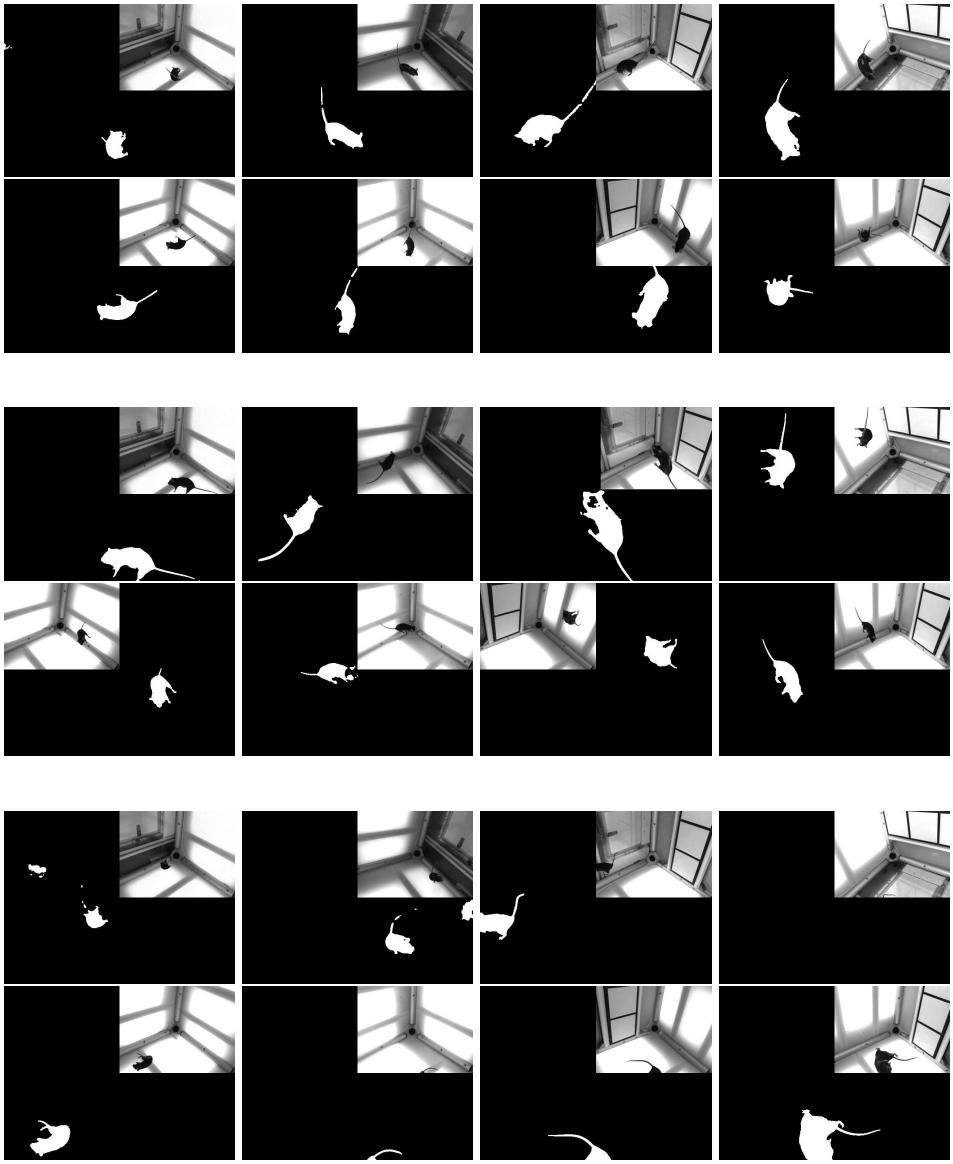


Figure 3.14: *Examples of background subtraction with the original image on the top right corner : top row shows a case where the results are satisfactory - middle row shows a case where the dark objective of the camera leads to false negative (see the head on camera 3 and 6) - bottom row shows a case where the absence of anti-reflexive material on the top panel leads to false positives (top left on camera 1). The backgrounds are those shown on Figure 3.13.*

edges of the polyhedron are computed by recursively using the geometry and orientation properties of the image contours. Finally, the edges are walked through to identify the faces.

Figure 3.17 shows the different reconstructions from the silhouettes displayed on Fig-

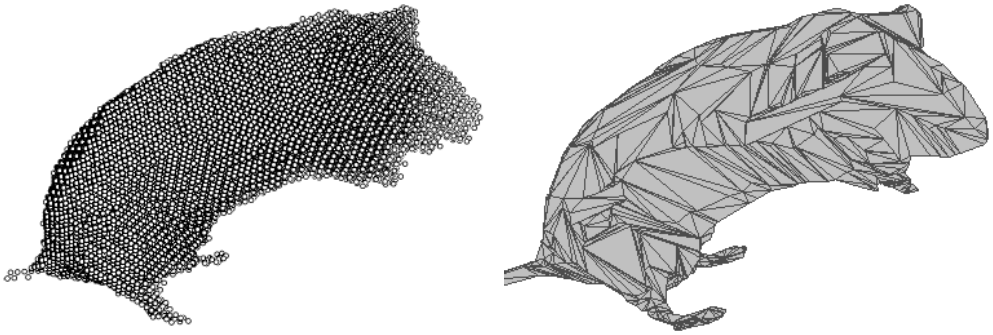


Figure 3.15: *Comparison of space carving and Exact Polyhedron Visual Hull using the silhouettes of the top row of Figure 3.14.*

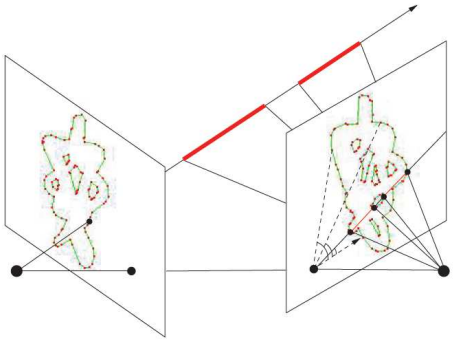


Figure 3.16: *From [FB03] - Viewing edges (red) along a given viewing line.*

ure 3.14. Obviously, the quality of reconstruction is highly dependent on the quality of the silhouettes as well as on the number of cameras. False negatives in the silhouettes lead to holes in the reconstruction (middle row of Figure 3.17). False positives can lead to small separated connexe components if there is no other view where it reprojects in the background. This is why the number of cameras that cover the mouse is important. If it is too low, there will not be enough information to discriminate along certain view directions which leads to a very coarse approximation of the visual hull (bottom row of Figure 3.17). When the quality of the reconstruction is good enough as in top row of Figure 3.17, we can expect to be able to measure differences in pose and motion between strains of mice. For instance, Figure 3.18 shows the EPVH of the fall posture the different strains tend to assume.

However, a comparison other than visual can not be done on these meshes as they are completely independent, we do not have correspondences between their vertices and the meshes do not even have the same topology. To quantify the differences between the different postures, we have to be able to match information in-between those

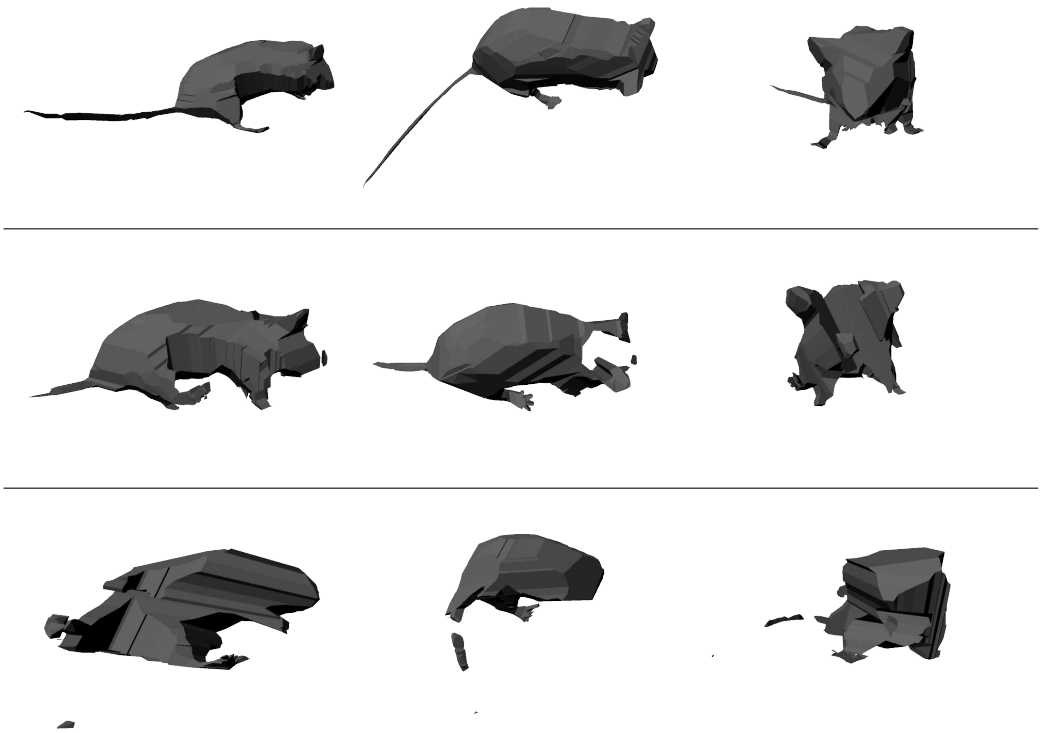


Figure 3.17: *Profile, top and front view of the Exact Polyhedron Visual Hulls (EPVH) reconstructed from the silhouettes on Figure 3.14. The quality of the visual hulls deteriorates with the quality of the silhouettes and the number of cameras that cover the mouse.*

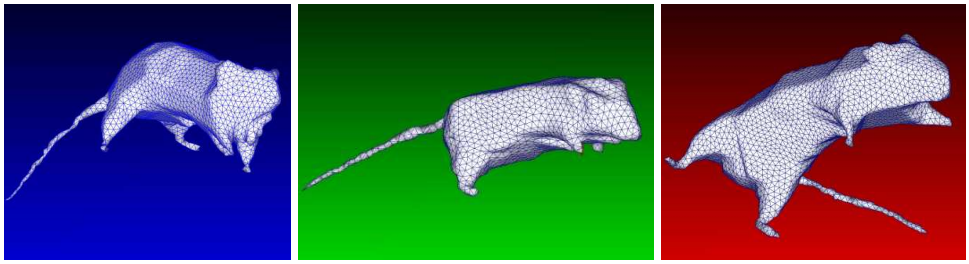


Figure 3.18: *From left to right : fall postures for control mice, ied mice and IsK mice (Section 1.2.1) - Control mice tend to adopt an adapted posture and maintain their spacial orientation, ied mice tend to adopt the same posture but do not seem able to maintain an appropriate spacial orientation, IsK mice do not seem to adopt any particular posture or orientation.*

meshes. For this, we need a model that we can deform to fit to the reconstructed meshes.

3.3 DEFORMATION MODEL OF THE ENVELOPE

Multi-view reconstruction applied to a sequence gives us a set of meshes with different topologies. As a result, extracting anatomical or motion measurements from this sequence of meshes requires to first identify on each and every one of these meshes where the features are located. Given the quality of the meshes (see Figure 3.17), this is a very difficult task as some of the features may not even be reconstructed (see left foot on bottom row of Figure 3.17). Another approach is to use an a-priori model of the observed object and deform this model to fit the observations. By doing so, any measurement defined on the model can be tracked through the sequence. The model can for instance be a result of multi-view reconstruction that comprehend all required features (for example, the reconstruction on the top row of Figure 3.17). It can also be manually modeled or be the result of a 3-D scan. In all those cases, we have the geometry and topology of the model but no way to deform it to fit the observations. To be able to do so, we need a deformation model. As the multi-view reconstruction technique used results in a mesh, we only consider meshes as models. First, Section 3.3.1 describes the deformation technique used. Then a parameterisation of this deformation in introduced in section 3.3.2.

3.3.1 Cage-based animation

As explained in section 3.1.2, rodents are not easy to animate because, unlike humans whose limbs are clearly defined, the limbs of rodents are sometimes not easily discernable from the body. For example, on Figure 3.19, whereas the thigh and the forearm are easy to identify on the left image, they can not be told apart from the trunk on the right image.

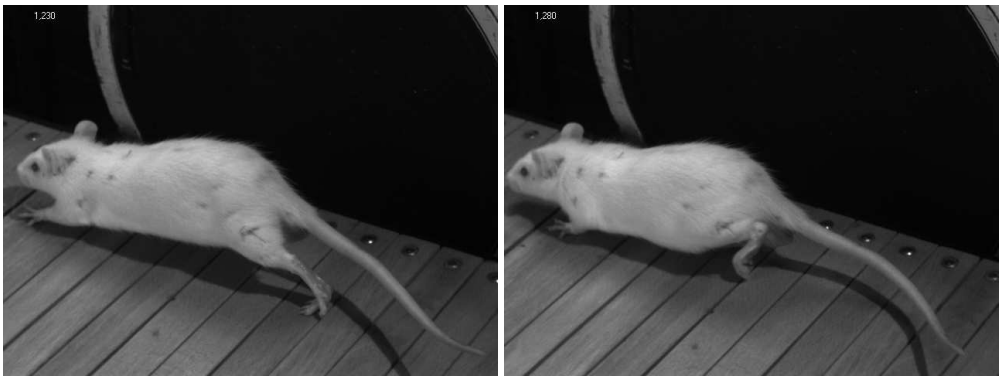


Figure 3.19: *Rat during a locomotion cycle.*

Due to this skin sliding effect, skinning does not seem an appropriate way to deform the model. Indeed, thanks to the tube-like shape of humans, an animation skeleton is quite straightforward to design for humans. However, to model deformations such as the ones on Figure 3.19, designing an animation skeleton and setting good skinning

weights is highly difficult. On top of that, an animation skeleton is a very constrained deformation model as it is usually designed to have at most 3 degrees of freedom per articulation in rotation and 0 in translation. Once again, this is very well suited for human beings but breaking these constraints becomes necessary to deform other models in a natural way. On the other hand, cage-based animation doesn't lead to such constraints as the cage that controls the deformations is a mesh. As a result, the degrees of freedom of the model are the positions of the vertices of the cage and those positions are unconstrained. We opt for a cage-based animation to deform the envelope as it has better properties in terms of shape preservation and smoothness than skinning for big deformations such as those that can be witnessed on rodents. From now on, the mesh to deform is called the model and the low-resolution mesh controlling the deformation is called the cage. As explained in Section 3.1.2, unlike other barycentric coordinates, Green coordinates are empirically proven to be quasi-conformal (i.e. the angles are preserved) in 3D and have a closed-form solution. We therefore choose these barycentric coordinates to generate the model from the cage.

To ensure that the Green Coordinates are quasi-conformal, the new positions of the vertices of the model are not only dependent on the positions of the vertices of the cage as expressed in Equation (3.5) but also on the orientation of the faces. Keeping the same notations as in Equation (3.5),

$$p_i = \sum_{j=0}^{m} \phi_j(p_i^0) v_j + \sum_{j=0}^{r} \psi_j(p_i^0) s_j n(t_j) \tag{3.6}$$

where p_i^0 is the position of the vertex i of the model prior to any deformation, $(t_0 \dots t_r)$ are the faces of the cage, $n(t_j)$ is the unitary normal of face t_j, s_j is the scaling factor associated with the face t_j and ϕ, ψ are the Green coordinates functions.

The Green coordinates functions ϕ and ψ are defined so that the deformation preserves the shape. Green's third identity states that for any harmonic function u in a domain $D \subset \mathbb{R}^d$, for any p in $interior(D)$:

$$u(p) = \int_{\partial D} (u(\xi) \frac{\partial G(\xi, p)}{\partial n(\xi)} - G(\xi, p) \frac{\partial u(\xi)}{\partial n(\xi)}) d\sigma_\xi \tag{3.7}$$

where G is the fundamental solution of the Laplace equation, n is the oriented outward normal to ∂D and $d\sigma_\xi$ is the area element on ∂D. Taking the domain enclosed by the cage to be D and defining $u(p) = p$, we can derive Equation (3.6) from Equation (3.7) with :

$$\phi_j(p) = \int_{\xi \in N(v_j)} \Gamma_j(\xi) \frac{\partial G(\xi, p)}{\partial n(\xi)} d\sigma_\xi \tag{3.8}$$

and

$$\psi_j(p) = - \int_{\xi \in t_j} G(\xi, p) d\sigma_\xi \tag{3.9}$$

where $N(v_j)$ is the union of all faces in the 1-ring neighbourhood of v_j, Γ_j is the piecewise-linear hat function defined on $N(v_j)$, which is 1 at v_j and 0 at all other vertices in the 1-ring and linear on each face. We refer to [LLCo08c] for these derivations as well as for the derivation of closed-form solutions in a 3-dimensional space.

The scaling factors s_j in Equation (3.6) are picked to ensure the following properties : linear reproduction, translation, rotation and scale invariance, shape preservation and smoothness. Linear reproduction ensures that applying Equation (3.6) without any deformation of the cage does not deform the model and is enforced by construction if $s_j = 1$ when the normal of face t_j does not change. Translation, rotation and scale invariance ensures that any affine transformation applied to the cage deforms the model as if the same affine transformation was directly applied to the model. Shape preservation ensures that the deformations are quasi-conformal in dimension 3. Smoothness ensures that ϕ_j and ψ_j are C^∞ inside the cage. In [LLCO08b], those conditions are used to define s_j such that, when the normal of tj changes, the deformation is the least deforming. 'The least deforming' is defined as introducing the less stretch on t_j. This is measured in 3D as some average chosen empirically between the singular values of the linear map deforming the face. This leads to :

$$s_j = \frac{\sqrt{\|u'\|^2 \|v\|^2 - 2(u'.v')(u.v) + \|v'\|^2 \|u\|^2}}{\sqrt{8} area(t_j)} \tag{3.10}$$

where (u, v) are vectors defining the edges in t_j prior to deformation and (u', v') are the vectors defining the same edges after deformation. This is the optimal coefficient to enforce shape preservation. However, s_j can be set to 1 for every face t_j if shape preservation is not required as in the example on Figure 3.28.

As a result, we have a closed-form solution for the Green coordinates and the scaling factors. To have fast interactive deformations, the coordinate functions are precalculated at the desired positions so that only the scaling factors (if required) and the sums in Equation (3.6) have to be computed to update the model when deforming the cage. There are no requirements on the model : the only information used is the position of its vertices. This is a very important as the models we aim at deforming are built from multi-view reconstruction and are far from being 2-manifold (see Figure 3.17).

The shape preserving property ensures that artifacts observed in linear blend skinning such as elbow collapse do not occur. Figure 3.20 shows both linear blend skinning and cage-based animation applied to same model. The animation skeleton used for skinning consists of two bones. The cage is built as a cube subdivided at the articulation between the two bones. It can be seen that the surface is better-preserved with cage-based animation as there is no elbow collapse effect at the articulation between the bones.

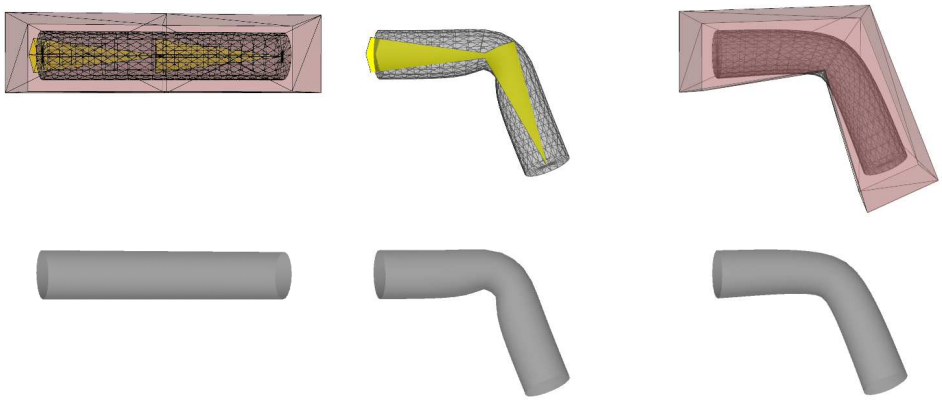


Figure 3.20: *Comparison between skinning and cage-based animation on a cylinder - left : initial model - middle : skinning - right : cage-based animation with Green Coordinates.*

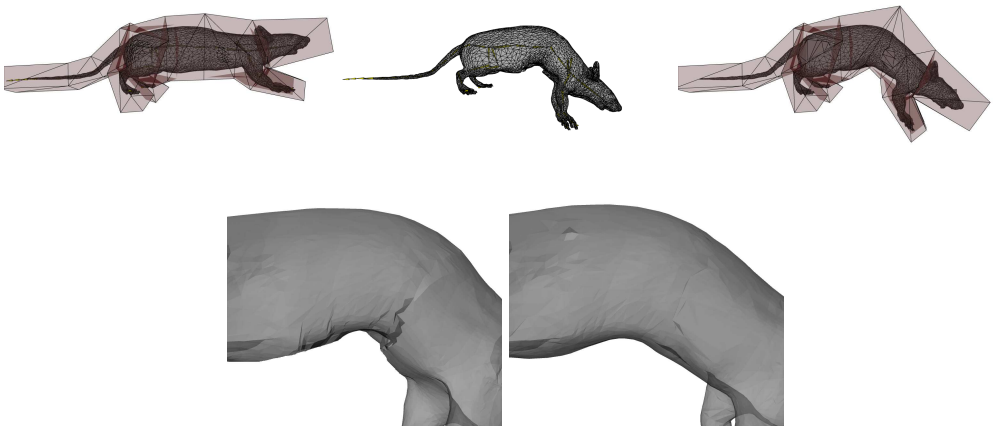


Figure 3.21: *Comparison between skinning and cage-based animation on a more complex model - top, from left to right : initial model, skinning, cage-based animation - bottom row, from left to right : zoom on skinning result, zoom on cage-based animation result.*

This, of course, can be observed on more complex models such as a rat on Figure 3.21.

Figure 3.22 shows an example of trying to use both skinning and cage-based deformation to fit a model made by an artist to an EPVH reconstruction from the parabolic flight dataset. As can be seen on the top row, the animation skeleton and the cage have the same rest pose on which both the Green coordinates and the skinning weights are computed. The skinning is a dual quaternion skinning with skinning weights computed automatically with Autodesk Maya [may]. Some manual tuning of the skinning

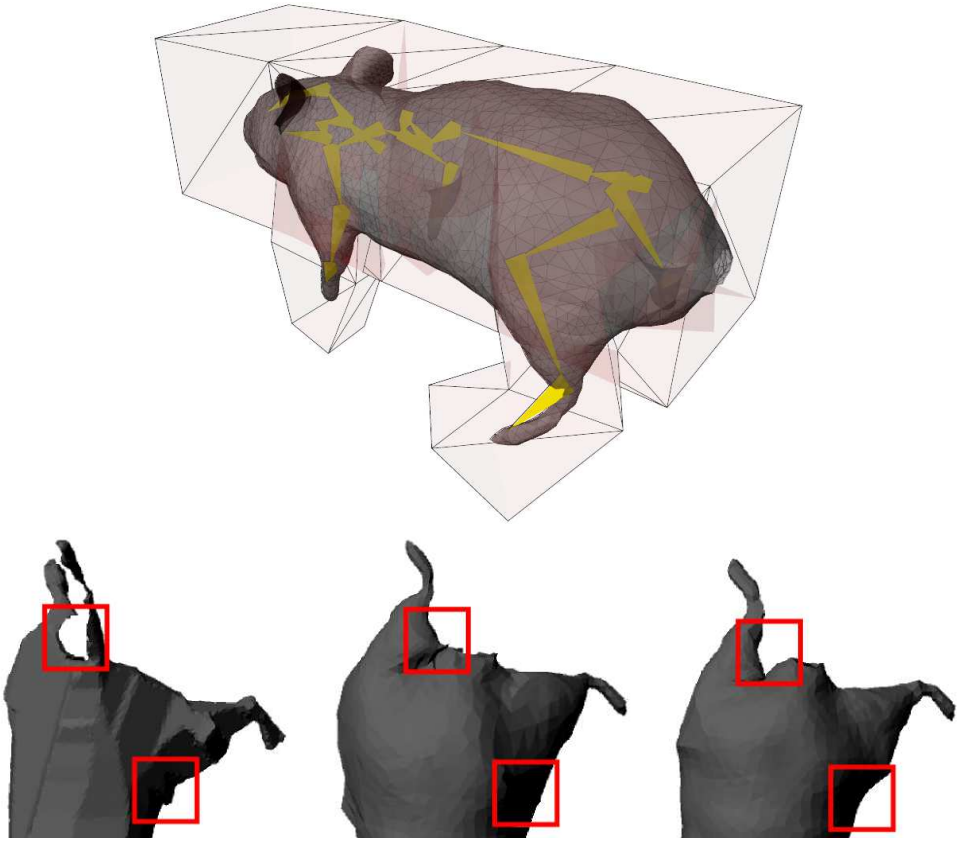


Figure 3.22: *Comparison between skinning and cage-based animation to fit a model from multi-view reconstruction - top row : animation skeleton (yellow), cage (red) and artist-made model (grey) in rest pose - bottom row from left to right : EPVH reconstruction, deformation by skinning, deformation by cage-based animation.*

weights had to be done to avoid some obvious ill-suited weights. On the bottom row, it can be noticed that the large deformation of the legs is a closer match to the EPVH for the cage-based animation whereas skinning results in a less clear separation between limbs and trunk (see between the rear of the animal and the right leg and between the abdomen and the left leg).

Another advantage of cage-based animation over skinning apart from smoother deformations is the fact that the Green Coordinates are C^∞ inside the boundary of the cage. As a result, given a certain cage, any model in a similar pose can be used or any modification in geometry or topology can be done to the current model and recalculating the Green Coordinates for the new vertices is done automatically. When using skinning, the weights often have to be adjusted manually by artists to result in the expected deformations. Therefore, any modification of the current model requires

tweaking once again the weights manually and using a different model either requires re-tuning the weights or using complex tools such as transfer functions. Figure 3.23 shows an example of using the same motion of the same cage to animate two very different humanoid models of the same size. No manual adjustment had to be made to transfer the animation from one model to the other.

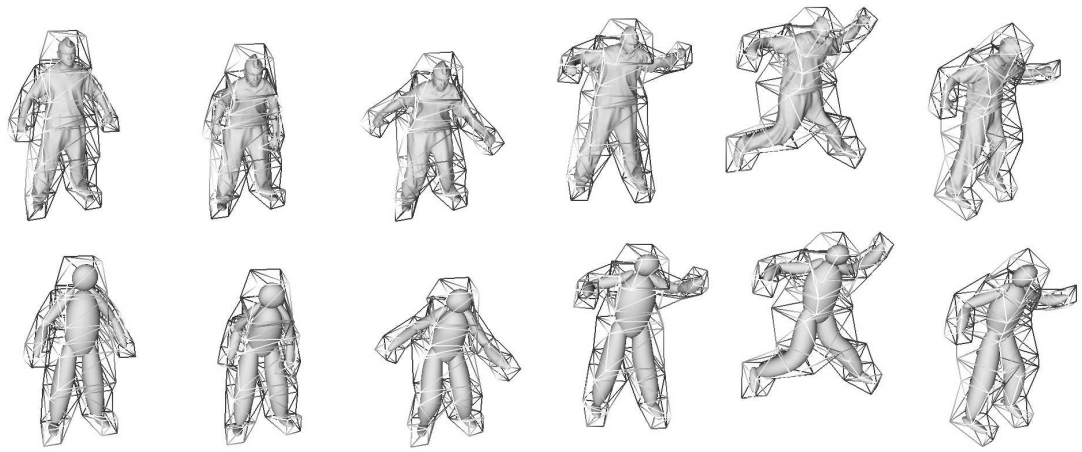


Figure 3.23: *Example of the same cage used to animate two different models.*

Cage-based animation is thus a flexible fully-automatic way of deforming meshes with ensured shape-preservation and smoothness properties. However, the degrees of freedom of a cage-based deformation model are the positions of the vertices of the cage. This has several drawbacks. First of all, this leads the cage to be counter-intuitive to create or deform. Indeed, whereas it is intuitive to create and animate a skeleton as each joint corresponds to an anatomical articulation, vertices of a cage do not represent a real-life object and can be moved unconstrained and independently from one-another in the 3D space. This lack of link to real-life features such as morphological structures also has the inconvenience of making the task of measuring or analysing motion from a cage-based animation hard. Indeed, when studying different gaits of animals from digitised animations, one can easily study the movement of the legs by studying the trajectories of the degrees of freedom of an animation skeleton [GBJD10]. However, there is no direct measure of those trajectories on a cage-based animation. This leads us to believe that, even though cage-based animation provides a better tool than skinning as a deformation model for rodents, it lacks a higher-level representation to make these deformations intuitive to design, manipulate and analyse. We develop such a representation in the next section.

3.3.2 Oriented Quads Rigging (OQR)

Being a coarse approximation of the model, the cage is easier to deform than the actual model but lacks the high-level representation that a skeleton conveys. Conversely to manipulating vertices, a higher-level representation leads to benefits in all steps of deformation modeling and analysing. Generating and deforming a cage becomes more intuitive and analysing these deformations becomes direct. For this, we introduce **Oriented Quads Rigging** (OQR).

Oriented Quads Rigging (OQR) provides an underlying structure to intuitively generate and deform the cage. Oriented quads have 8 degrees of freedom : 2 in scale, 3 in translation and 3 in rotation, defined globally. Their geometric representation is a rectangle. This quad is defined from a canonical quad transformed according to the degrees of freedom. The canonical quad is defined as a square of size 1 centered on $(0, 0, 0)$ whose normal is $(0, 0, 1)$ and whose rotation around its normal, the Z axis ,is null. That means that the four vertices defining the canonical quad have coordinates :

$$\begin{aligned} c_0 &= (-0.5, +0.5, 0) \\ c_1 &= (+0.5, +0.5, 0) \\ c_2 &= (+0.5, -0.5, 0) \\ c_3 &= (-0.5, -0.5, 0) \end{aligned} \tag{3.11}$$

The degrees of freedom of the oriented quads are :

- two degrees of freedom in scale $S = (s_x, s_y)$;
- three degrees of freedom in rotation θ that define a rotation in 3D. We call R_θ the rotation representation of this rotation ;
- three degrees of freedom in translation T that define a translation in 3D.

With those notations, the vertices of a quad are computed as :

$$\begin{aligned} c_0(S, \theta, T) &= R_\theta \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} c_0 + T \\ c_1(S, \theta, T) &= R_\theta \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} c_1 + T \\ c_2(S, \theta, T) &= R_\theta \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} c_2 + T \\ c_3(S, \theta, T) &= R_\theta \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} c_3 + T \end{aligned} \tag{3.12}$$

which can be written using the Rodrigues rotation formula :

$$c_i(S,\theta,T) = cos(\alpha)c_i(S) + (e \times c_i(S))sin(\alpha) + e(e.c_i(S))(1 - cos(\alpha)) + T \quad (3.13)$$

where e is the unit vector of the axis of the rotation R_θ and α its angle of rotation and :

$$c_i(S) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} c_i \quad (3.14)$$

Figure 3.24 shows the effect of each degree of freedom on the canonical quad. We

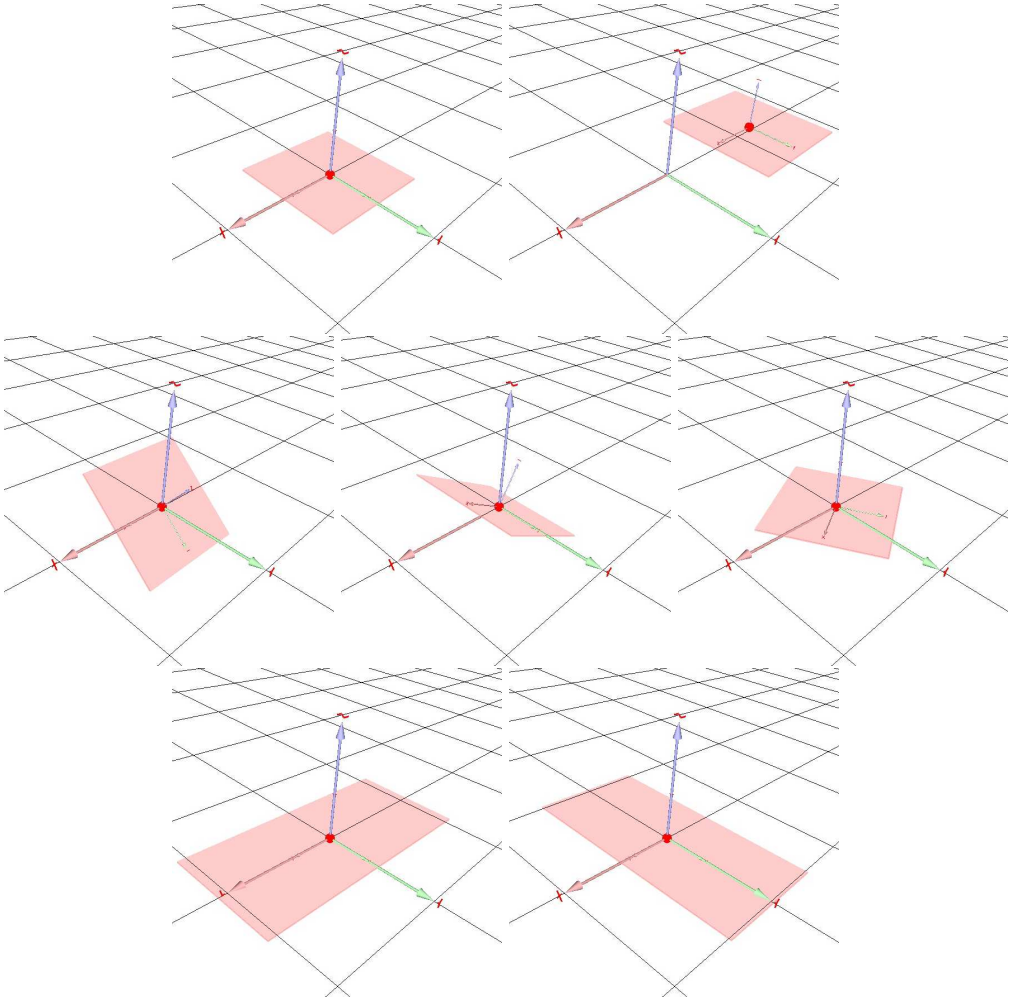


Figure 3.24: *Different degrees of freedom of an oriented quad - Top row : canonical quad and canonical quad translated - Middle row : rotation about the two directions defined by the quad and rotation about the normal - Bottom row : scale along the two directions defined by the quad.*

can therefore animate those oriented quads by modifying their 8 degrees of freedom.

We can modify in the world space their translation and orientation as well as the scale in both directions X and Y. For their orientation, a very intuitive way to interactively modify the orientation is to use as degrees of freedom the rotation about the two directions defined by the perpendicular edges of the quad to set the normal of the plane and the rotation about that normal to set the 2D orientation in the plane supporting the quad.

A set of r oriented quads is then used as the underlying structure of a cage. We call this structure an **Oriented Quads Rigging** (OQR). In the following, we call c_k^l, $k = 0..3$, $l = 0..r-1$ the k^{th} vertex (as defined in Equation (3.11)) defining the geometry of the oriented quad l. The topology of the cage is defined from the set of oriented quads by

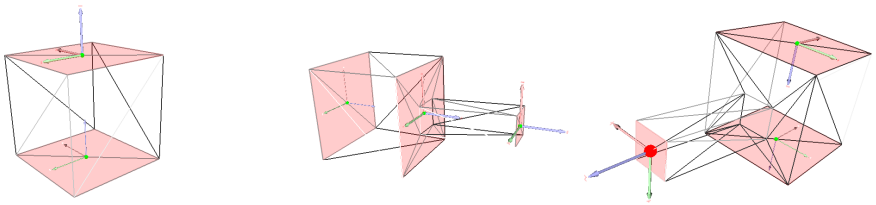


Figure 3.25: *Three types of relation that can exist between oriented quads - from left to right : bridge, extrusion, fork. The oriented quads are displayed in red and the resulting cage is displayed in black wireframe.*

a set of relations in-between those oriented quads. We identify three types of relation (all represented in Figure 3.25) :

- **bridge** : Two oriented quads l_0 and l_1 can be linked (see left of Figure 3.25). To do so, we create four faces that link the vertices $c_k^{l_0}$ to the vertices $c_k^{l_1}$ to build a parallelepiped with the geometry of the oriented quads as opposite faces.
- **extrusion** : We create a branch from a quad contained inside the oriented quad l_0 to the oriented quad l_1 (see middle of Figure 3.25). To do so we create four new vertices $\tilde{c_k^{l_0}}$ inside quad l_0. Then, we mesh l_0 to create a hole defined by the vertices $\tilde{c_k^{l_0}}$, thus creating 8 triangles. We then use a bridge between the vertices of the hole and l_1. Figure 3.27 illustrates the process.
- **fork** : We create a branch from one of the faces of a parallelepiped built by a bridge to an oriented quad l_1 (see right of Figure 3.25). The process is the same as with an extrusion.

In the end, the faces t_j of the cage (see Equation (3.6)) are the faces created by applying bridges, extrusions and forks plus the oriented quads themselves if they have not been meshed by an extrusion and if they are used in at most one bridge. If they are

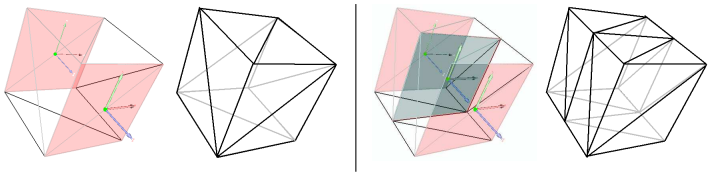


Figure 3.26: *When a quad is used in only one bridge (left), its geometry becomes a face of the cage. When it is used in two bridges as is the middle quad on the right, its face does not contribute as a face of the cage but its vertices are used to build other faces.*

used in two bridges then they are inside the cage and a face must not be created, as illustrated in Figure 3.26.

These relations give us the topology of the cage, which remains the same during cage-based animation. However, the positions of the vertices of the cage v_j (see Equation (3.6)) have to be updated procedurally from the vertices c_k^l when the degrees of freedom of the oriented quads are changed. We differentiate two cases :

- the vertex v_j is one of the c_k^l. It is a vertex of an oriented quad (for example, the vertices v_0, v_1, v_2 and v_3 in Figure 3.27). It is therefore built from the translation, rotation and scale of the oriented quad l using Equation (3.12). We call the set of vertices built this way V_c.

- the vertex v_j is one of the $\tilde{c_k^l}$. It is not a vertex of an oriented quad : it was created by applying an extrusion or a fork (for example,the vertex v in Figure 3.27). We call the set of vertices built this way V_b. These vertices are defined by barycentric coordinates i.e. for each vertex v in V_b :

$$v = v_0 + \alpha(v_1 - v_0) + \beta(v_3 - v_0),\ v_0, v_1, v_3 \in V_c \tag{3.15}$$

  Therefore, once all the vertices of V_b have been updated, the vertices of V_c can be updated. Figure 3.27 illustrates α and β.

As a result, each vertex of the cage either belongs to V_c and therefore is defined by the global position and orientation of one quad or it belongs to V_b and therefore is defined by linear interpolation of 3 vertices that belong to V_c. The cage can thus be built automatically from the oriented quads in a procedural manner. Deforming the model then consists in setting the translation, orientation and scale of each quad, which procedurally defines the vertices of the cage through Equations (3.12) or (3.15). Finally, this cage defines the vertices of the model by linear combination of the vertices of the cage and its normals through Equation (3.6).

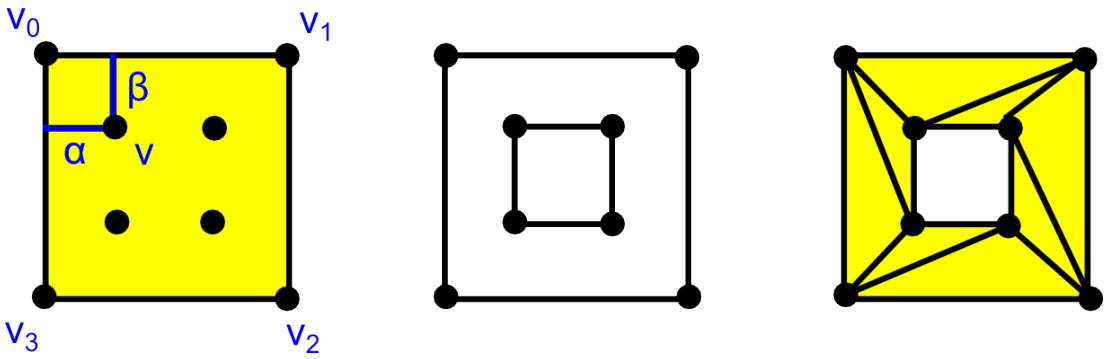


Figure 3.27: *To create a branch, in case of an extrusion or a fork, first vertices are defined on the quad where the branch is created (left) in order to create a hole (center) then the face is re-meshed around the hole (right).*

We have thus described an intuitive way of generating and manipulating cages for cage-based animation. It makes deformations that are most of the time undesirable such as fold-overs highly unlikely to happen whereas they can easily happen when manipulating vertices independently. To sum it up, OQR combines the high-level representation of skeletons with the smooth and shape-preserving deformations of cage-based animation.

The main advantage of OQR over animation skeletons is that the degrees of freedom of the oriented quads are unconstrained. The oriented quads are defined globally and can move in translation, orientation and even scale whereas animation skeletons are often represented as a hierarchy of ball-and-socket joints. Indeed, animation by oriented quads goes beyond articulated motion and can model a wide variety of movements. For example, Figure 3.28 shows how 2 oriented quads can be used to animate the traditional squash and stretch effect, which is one of the 12 basic principles of animation as defined by Walt Disney. By keyframing the 2 quads on the five frames highlighted on the top left image of Figure 3.28 and interpolating in-between those frames (top right of Figure 3.28), we apply the procedural construction of the cages as described before (middle left of Figure 3.28) and compute the cage-based deformation resulting on the ball (middle right of Figure 3.28). It should be noted that this is a real-time process as the update of the vertices of the cage corresponds to either an affine transformation if the vertex belongs to V_c (Equation (3.12)) or a linear combination if it belongs to V_b (Equation (3.15)). As for updating the model, it is also a linear combination (Equation (3.6)) and is therefore $O(nm)$ where n is the number of vertices of the model and m is is the number of vertices of the cage. As the number of vertices of the cage m is by construction far lower than the number of vertices of the model n, updating the model is $O(n)$.

Of course, one other advantage of oriented quads is that it provides a standard morphological representation for articulated objects that would usually be animated with

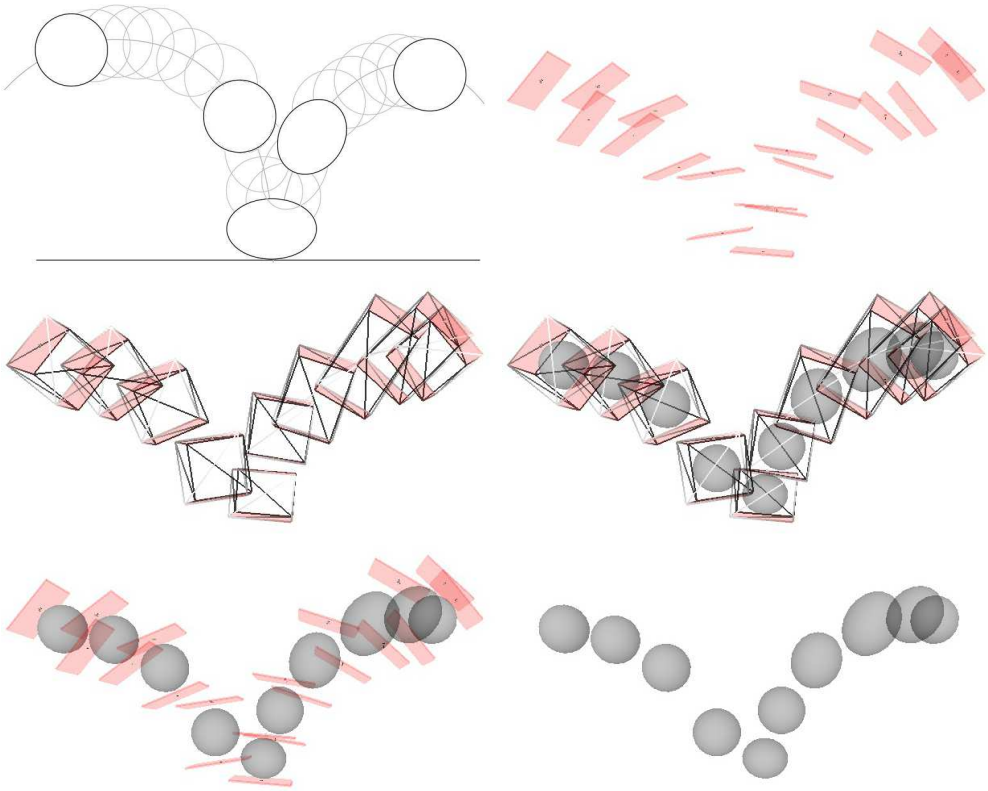


Figure 3.28: *Example of how oriented quads can be used to create a squash and stretch effect with a few keyframes.*

skeletons. Indeed, by positioning the oriented quads at the interface between rigid zones of the model, typically where joints would be located in joint-based animation, we can define a cage that has the same morphological meaningfulness as joints. For example, an OQR of 20 oriented quads is used to model rodents (see section 3.4 for more details). This cage can then be used to animate either a rat (Figure 3.29) or a mouse (Figure 3.30). Notice on the bottom row of Figure 3.30 how we can model both full extension and full folding of the mouse with only 4 quads on the spine. If we were using animation skeletons, creating such a large bend with a smooth deformation of the skin would require much more than 4 joints on the spine. Indeed, the constant length of bones in animation skeletons leads to the use of many joints to be able to model highly deformable objects such as the spine of rodents.

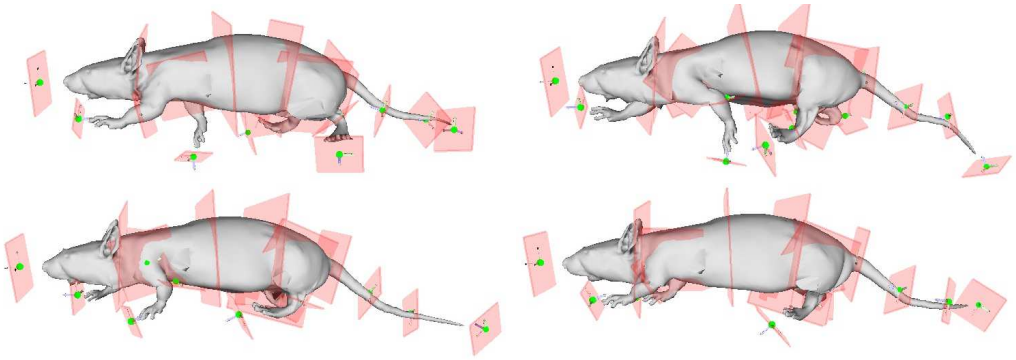


Figure 3.29: *Example of how oriented quads can be used to create an animation of an articulated objet, here the locomotion of a rat.*

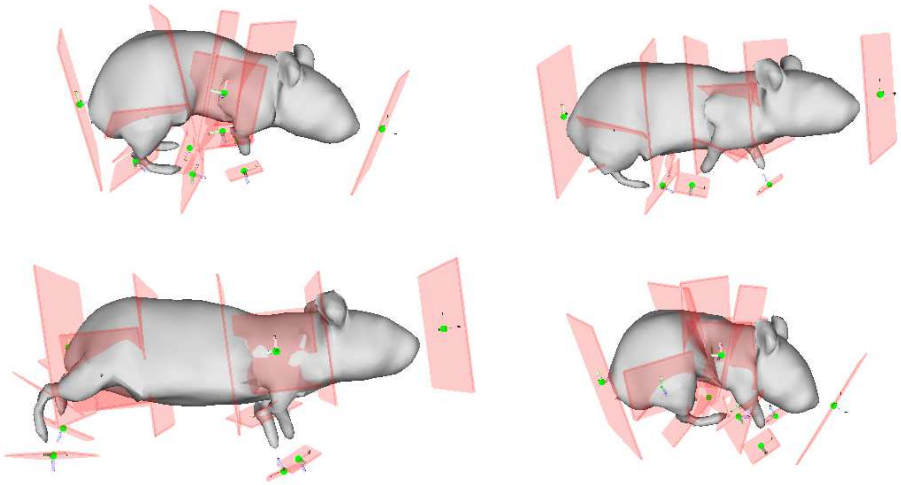


Figure 3.30: *Example of how oriented quads can be used to create an animation of an articulated objet, here different postures of a mouse.*

3.4 LINK TO SKELETAL STRUCTURES

We have thus introduced Oriented Quads Rigging (OQR), a flexible high-level rigging that allows for smooth shape-preserving deformations. In the case of rodents, the OQR is built from 20 quads. Figure 3.31 shows the OQR used for rats and mice. It can be seen super-imposed with the skeletal structures on Figure 3.33. The head is bordered by 2 quads. The spine is rigged with 4 quads : one at the first cervical vertebra, one at the last thoracic vertebra, one at the last lumbar vertebra and one at the fourth caudal vertebra. The tail is rigged with 4 quads. Due to the amount of soft tissue on rodents, the quads rigging the legs do not correspond to skeletal structures but to places where the skin folds. Both forelegs and hindlegs are rigged with 3 quads each. From these models, we also build a tail-less model made of 17 quads both for the rat and the mouse (right of Figure 3.31). Such models allow us to animate the envelope of rodents through a high-level representation in a temporally coherent manner.

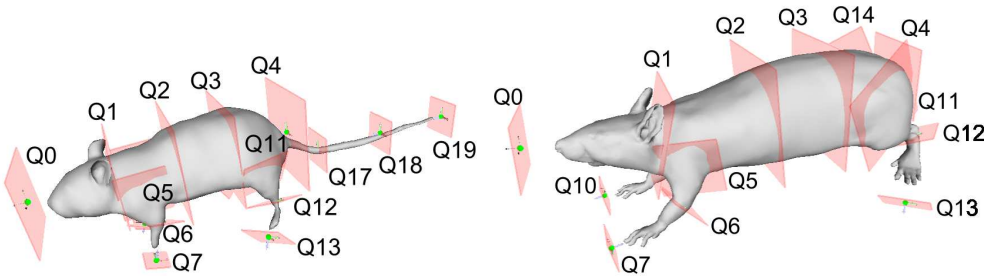


Figure 3.31: *Oriented Quads Rigging (OQR) for rodents - top : mouse, bottom : rat without the tail.*

However, measures on the envelope are not what we aim at extracting. As explained in Chapter 2, accurate measure of internal structures is paramount. In the same chapter, we developed a model that allows us to track the bones through weighted Inverse Kinematics thanks to markers. This model was validated for tracking with external markers on X-ray videos. The results show that, even though the precision of the tracking is dependent on the number of markers, precise reconstruction of the motion of the bones can be achieved with a reduced number of markers in comparison to SVD-based methods. There are two main reasons for that : the use of anatomical constraints through angular springs that attract the solids towards their rest pose and the use of different weights for the different markers, which conveys a confidence in the relevance of the different markers.

Consequently, we can expect to track the bones with accuracy by simulating exter-

nal markers with the OQR. The markers can be any point within the cage defined by the OQR. Indeed, any point within the cage is animated when the degrees of freedom of the OQR are changed. As a result, the animation trajectory of this point as reconstructed through linear interpolation thanks to its Green Coordinates can be used as a trajectory of a marker as in Section 2.4. A first idea is to directly use the degrees of freedom of the cage. This leads to using the positions of the quads of the OQR as markers. However, due to the low resolution of the rigging, this leads to an under-constrained tracking as can be seen on left of Figure 3.32.

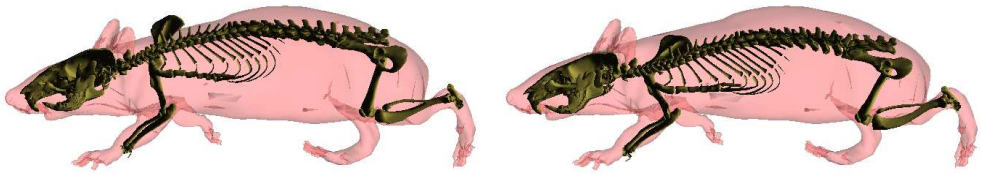


Figure 3.32: *Tracking of the bones using the position of the quads of the OQR (left) and vertices of the skin (right) as markers. Note how the problem is under-constrained when using quads, which leads to the scapula being improperly tracked.*

As the OQR has enough quads to model the deformations we study, instead of increasing the number of quads to have a more constrained system, we decide to use points within the cage as markers. There are several options to select those points. For each solid or articulation, the point on the skin model that is closest to it can be automatically selected. However, the solution that yields the best results is to use marker-based studies such as ratC6X1M1lo or, even better if available, ratC0X2M1lo to define a set of markers that are necessary and sufficient to achieve a given precision and accuracy. Once such a study is done and a set of points is selected, they can be used for all following studies. As can be seen on Figure 3.32, this approach gives better results than using the centers of the quads.

In our case, inspired by our experience on sequence ratC6X1M1lo, we define a set of 27 markers that can be seen on Figure 3.33 for the left side of the animal. We select 3 markers on anatomical points of the head (tip of the nose and ears) so that the position of the skull is completely defined by those three points. We select 4 markers on the spine, 6 markers on each foreleg and 4 markers on each hindleg. Figure 3.34 shows results of Inverse Kinematics for the sequence created on Figure 3.29. Thanks to the 3 markers on the head, the skull is always properly tracked. On the bottom right image, one can see that the radius is outside the skin of the forearm. This is due to the fact that the position of the forearm set manually is not reachable by the internal structures, except by violating anatomical constraints. It can be seen on other images as well that sometimes the elbow is slightly out of the skin. This is due to the fact that we have chosen the resolution of the cage to be quite low. Indeed, out goal is to have a compact representation that conveys information about the shape, not a perfect skin

model. And, as can be seen on Figure 3.34, even though the shape of the forearm is not always detailed enough to model the elbow correctly, it does not impede the correct reconstruction of the radius.

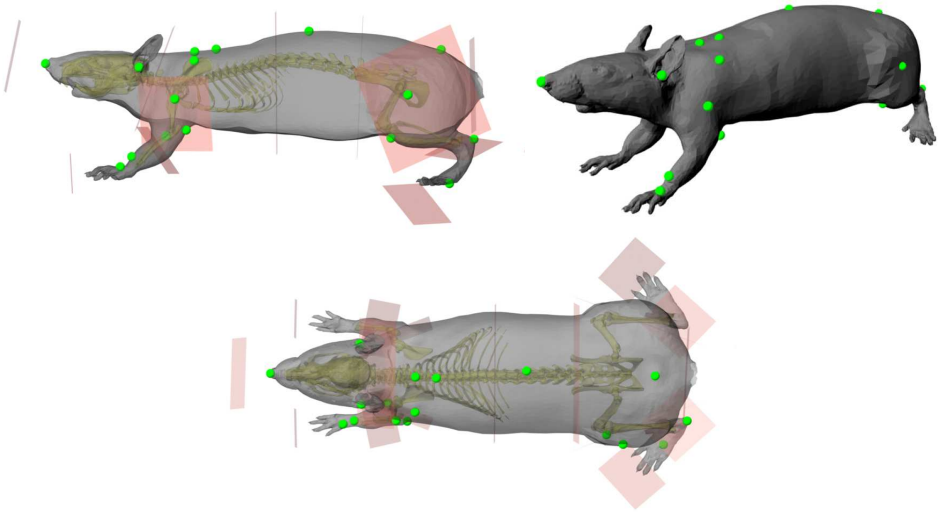


Figure 3.33: *Set of skin points (green spheres) selected to be used as markers for the tracking of internal structures from OQR configurations.*

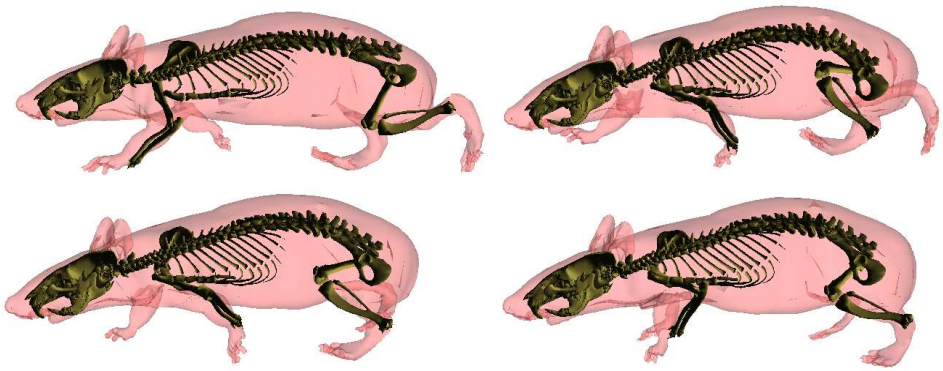


Figure 3.34: *Examples of tracking of the bones during locomotion using the OQR of Figure 3.31 and the markers of Figure 3.33.*

3.5 CONCLUSION

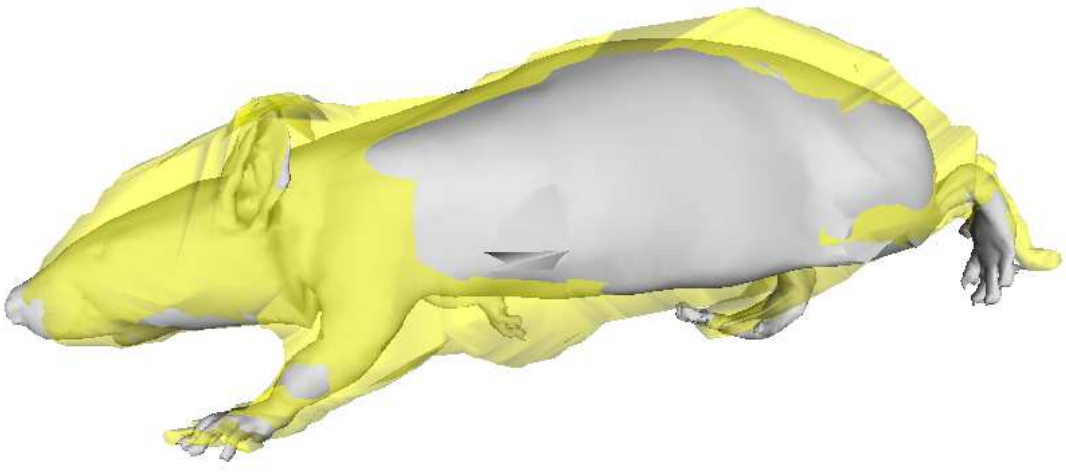
We have proposed a new animation primitive, the Oriented Quads Rigging (OQR) as an alternative to skeleton-based animation. OQR has a high-level representation that enables the extraction of meaningful motion parameters directly from its low-dimensional representation. It also allows for smoother and more flexible deformations compared to skinning as it is a layer on top of cage-based animation. Even though we use it to animate articulated creatures, it is a very general primitive that can be used in many contexts. In the context of our studies, it allows for realistic and coherent trajectories of the vertices of the deformed mesh when animating the model. As a result, a sub-set of those vertices can be used for the weighted Inverse Kinematics framework derived in Chapter 2 to recover the skeletal structures from the OQR configuration.

With both Chapter 2 and Chapter 3, we have built a complete rodent model that integrates both the skeleton and the skin of the animal and is animated by a high-level structure, the OQR. Consequently, if we are able to track the OQR configuration from the videos, we can extract the trajectories of the skeletal structures.

CHAPTER

4

MANIFOLD LEARNING FOR ORIENTED QUADS RIGGING RECOVERY.



So far, we have proposed and validated an animation model for both the envelope and the skeleton of rodents. Given a set of oriented quads organised as an Oriented Quads Rigging (OQR), we can deform the mesh representing the envelope of the animal. Vertices of this mesh can then be used as markers in weigthed Inverse Kinematics to animate the internal structures. To be able to track the bones from markerless videos, we propose a model-based approach for the recovery of the parameters of the OQR from multi-view set-ups. To regularise the tracking, the OQR space is constrained to plausible configurations using manifold learning (Section 4.2). The motion recovery can be done from a sequence of 3D meshes without temporal coherence extracted from the multi-view set-up as in Section 3.2 (Section 4.3) or directly from the images (Section 4.4).

4.1 Previous work

4.1.1 Motion recovery from a sequence of meshes

Motion recovery using multiple view environments has known important developments in both computer vision and computer graphics. Applications to biomechanics and character animation require a reliable and interpretable measurement of motion. In that respect, a recovery of parameterised motion appears as a relevant goal. As a preprocess, it is now common to calibrate multiple views to obtain a streams of 3D meshes from visual inputs such as silhouettes as explained in Section 3.1.1. The resulting information from these methods is not temporally structured nor related to interpretable parameters in terms of motion.

Skeleton-based methods impose a strong prior on how the surface deforms. These approaches have shown successful results such as in [VBMP08] [UFF06] [GSdA+09]. In those cases, a skeleton of a human or animal character has to be aligned on a reference template mesh of high quality. A skinning algorithm deforms the mesh, which is matched against 2D silhouettes. However, as explained in Section 3, the skinning algorithm is known to be sensitive to a careful tuning of vertices weights, which are often manually adjusted after an automatic computation, and can not model correctly the type of deformations one can observe on the envelopes of rodents. On top of that, a skeleton prior can sometimes be too constraining. For instance, if the knee is not properly registered, the fixed length of the leg makes it impossible to properly fit the foot, leading to a completely erroneous result.

On the opposite direction, feature-level methods such as the patch-based approach of [CBI10] offer a maximum flexibility. This approach, where vertices are grouped in rigid patches, is robust to large changes in the topology of the analysed scene as no assumption is made on the internal structure. However, it does not fulfill the objective of motion recovery as no interpretable parametrisation can be directly derived from this formulation. Furthermore, in case of very noisy input meshes, an approach that

works at this level of locality risks providing unreliable results with non recoverable drift.

In contrast to existing methods for mesh tracking, we propose to relax the strong parametric constraint of the skeleton and deliver a more structure output than patches by making use of the Oriented Quads Rigging (Section 3.3.2). Contrary to skinning weights, the weights provided automatically by Green Coordinates show superior stability and do not require further tuning. The output is thus a sequence of values for the degrees of freedom of the OQR that can directly be used to interpret motion or for generating a sequence of temporally coherent coordinates of the vertices of the model, which can then be used for weighted Inverse Kinematics. Tracking mesh data using a cage for barycentric coordinates has first been proposed by [SF10] and recently improved by [TTB12]. However, those works assume temporal coherence of the input 3D data, which is not the case in our inputs.

However, for most methods, the accuracy of tracking may still be challenged by lighting conditions, poor camera positioning and occlusions, especially when studying animals and not actors in a capture studio. For instance, [VBMP08] allows the user to intervene when ambiguity leads to an incorrect fit of the skeleton (Figure 4.1). One of the key factor is therefore the ability to bring an efficient regularisation to the tracking problem. In [SF10], the cage deformation is constrained by a Laplacian smoothing. It has been reported that it did not prevent the cage from collapsing when tracking long sequences. For a better regularisation, we constrain the OQR to plausible configurations using manifold learning.

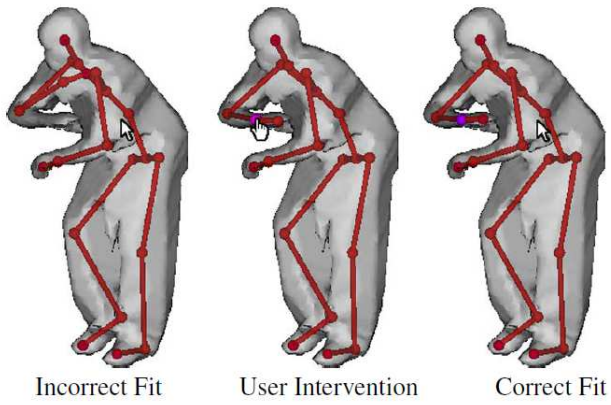


Figure 4.1: *User intervention when the skeleton is incorrectly fitted to the silhouettes in [VBMP08].*

4.1.2 Manifold learning for tracking

The success of manifold learning for robust tracking of human motion from monocular videos has been typically proven with works such as [UFF06]. For animation, an application of manifold learning has also been shown for 3D pose editing by [GMHP04]. We use it in the context of motion recovery as a postprocess of 3D reconstruction of meshes from multiple view silhouettes. We build on the formalism introduced in [PR11a] and [PR11b] for a rigorous derivation of the gradients of the object measurement with respect to the learned space.

While [PR11a] uses manifold learning of 2D contours to segment 2D images, we use a non-linear manifold as a deformation prior to track OQR configurations. We use GP-LVM to learn a space of plausible OQR configurations (Section 4.2). This allows us to constrain the search space for motion recovery from a sequence of meshes. (Section 4.3). In the prospects of using other cues such as appearance, we also derive a tracking algorithm directly from the images (Section 4.4).

4.2 MANIFOLD LEARNING OF OQR

Even though the Oriented Quads Rigging somewhat regularises the tracking problem by preventing unrealistic deformations such as the shrinking of the circumference of a limb if the scales of the quads are fixed, nothing preserves properties such as the length of a limb or prevents non physiological configurations. The learning of a non-linear manifold of the possible positions and orientations of the quads enforces such constraints. We learn three different manifolds as Gaussian Process Latent Variable Models : one for locomotion, one for open field exploration and one for parabolic flights.

As in [PR11a], where GP-LVM are used to segment 2D images, we use GP-LVM to learn a manifold of OQR configuration. A *Gaussian Process Latent Variable Model* (GP-LVM) is a low dimensional representation of the original data called the latent space. Each data point y_i of dimension d can be represented by a latent point x_i of dimension q, where q is smaller than d. Conversely, every point x in the latent space projects to a data point through a Gaussian distribution defined by its mean μ and its variance σ^2.

As derived in [Law05], given n learning samples $Y = [y_1 \dots y_n]$, those learning samples are mapped to n latent points $X = [x_1 \dots x_n]$ such that :

$$P(Y|X,\theta) = \prod_{i=1}^{n} \mathcal{N}(y_i|0, K(\theta)) \tag{4.1}$$

where $K_{i,j} = k(x_i, x_j)$, k being the covariance function defined by the hyper-parameters θ. For instance, if k is a radial basis function, as in our case, it is computed as :

$$k(x_i, x_j) = \alpha exp(-\frac{\gamma}{2}(x_i - x_j)^T(x_i - x_j)) \tag{4.2}$$

and the hyper-parameters are $\alpha \in \mathbb{R}$ and $\gamma \in \mathbb{R}$.

To learn the GP-LVM, Equation (4.1) is maximised with respect to both X and θ. That means that, given n learning samples Y and a latent space dimension q, we search for X of size $n \times q$ and θ that maximise the likelihood of Y given X and θ. Once X and θ are learned, we know that every point $x \in \mathbb{R}^q$ in the latent space projects back to a data point through a Gaussian distribution defined by its mean $\mu \in \mathbb{R}^d$:

$$\mu = Y^T K^{-1} K_x \tag{4.3}$$

where K_x is the covariance between x and all the learned latent points and its variance $\sigma^2 \in \mathbb{R}$:

$$\sigma^2 = k(x,x) - K_x^T K^{-1} K_x \tag{4.4}$$

However, there is no way to map a point $y \in \mathbb{R}^d$ in the data space to a point $x \in \mathbb{R}^q$ in latent space. GP-LVM gives a smooth mapping from latent to data space. That means that points close in the latent space will be close in data space. However, the converse is not true : GP-LVM does not guarantee that points close in the data space are close in the latent space. To constrain points close in the data space to be close in the latent space, back-constraints have to be imposed [LQnC06]. To do so, the latent points $(x_i)_{i=1..n}$ are made a smooth function of the data :

$$x_{i,j} = f_j(y_i, w) \tag{4.5}$$

where w are then the parameters we optimise for instead of x_i. By constraining the latent points to be a smooth mapping from data space, we enforce that small distances in data space are small in latent space. Usually, f_j is a kernel based function such as a radial basis function :

$$f_j(y_i, w) = \sum_{m=1..n} w_{j,m} exp(-\frac{\gamma}{2}(y_i, y_m)^T(y_i, y_m)) \tag{4.6}$$

where γ controls the smoothness of the mapping i.e. the closeness in latent space.

In our case, each data point y_i describes one deformation by the positions and orientations of the r quads in the OQR. Following the notations of Equation (3.12), we have :

$$y_i = [\theta_0 T_0 \dots \theta_r T_r] \tag{4.7}$$

where θ_j (respectively T_j) is the rotation (respectively the translation) of the quad j of the OQR. The scales of the quads are considered fixed, which is a safe assumption for articulated creatures such as human and animals.

In the case of rodents, the learning samples y_i are built manually by having the user locate the quads on EPVH extracted as in 3.2. We will show in Section 4.3.3 how the learning samples can be built automatically when motion capture databases are available, as is the case for humans. Prior to building Y from the learning samples, we align all the samples with respect to the position and orientation of the abdomen (Q2 in Figure 3.31). We then learn the latent variables and hyper-parameters by maximising Equation (4.1) using back-constraints.

Figure 4.2 and Figure 4.3 show two examples of latent space. Figure 4.2 shows a latent space of OQR for rat locomotion built from 6 learning samples (blue points). Figure 4.3 shows a latent space of OQR for the parabolic flights of mice built from 8 learning samples (blue points). The grey level value of a point in the latent space corresponds to the variance at that given point in the latent space. The whiter the pixel, the

higher the variance, the more likely the point is to correspond to a plausible configuration of the OQR. Red points and their corresponding reconstructions as computed by Equation (4.3) show that the latent space is suitable for interpolation between learned configurations. All those shapes are new shapes that are physiologically reachable.

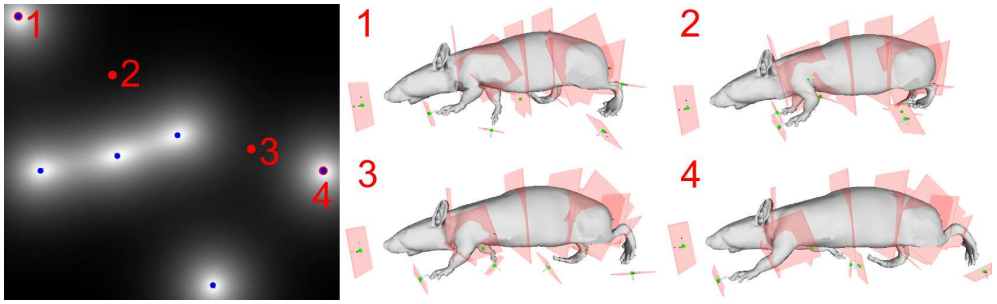


Figure 4.2: *Example of a 2D latent space of OQR and corresponding deformations for rat locomotion.*

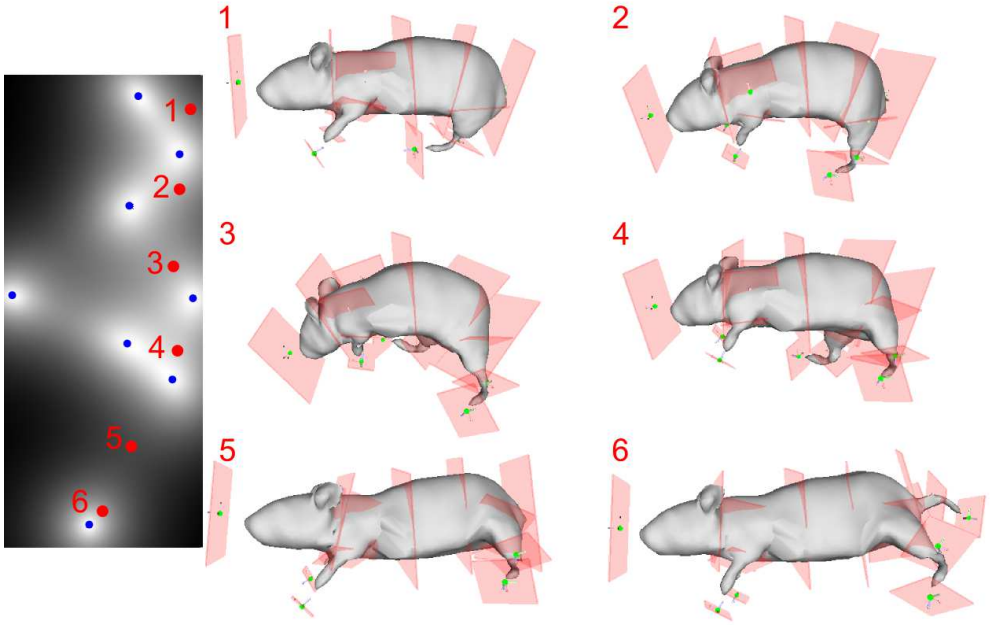


Figure 4.3: *Example of a 2D latent space of OQR and corresponding deformations for parabolic flights.*

4.3 MESH TRACKING

To recover the motion of the OQR from videos, we must fit the *model*, i.e. the shape being deformed by the OQR, to the current representation of the shape, namely the *target*. In our case, the *target* is a 3D mesh, the visual hull reconstructed from multiple views. To do so, we derive a differentiable energy that measures how well the model fits the target (Section 4.3.1). Instead of searching in the space of all OQR configurations, we differentiate this energy in the latent space to recover the deformation of the model (Section 4.3.2). By doing so, we stay in the learned manifold i.e. in the space of plausible shapes.

4.3.1 Objective function

In deriving an energy that measures how well the model fits the target, we can not make assumptions on the temporal coherence of the target meshes. As the input is a stream of independent 3D meshes with each their own geometry and topology, which is different from the geometry and topology of the model, the energy can not rely on correspondences between vertices as in [FB03]. To measure how well the model mesh fits the target mesh, we thus need to measure the distance between two different meshes having potentially two different topologies. As we will minimize this distance, we need a function which is differentiable with respect to the model mesh.

To that end, one can first estimate vertex-to-vertex sparse correspondences, then use those pairs of vertices to compute the distance between the two meshes. However, computing correspondences is costly and can fail, especially with data as noisy as ours. Another approach is to directly use a mesh to mesh distance, such as the Hausdorff distance. Even though the Hausdorff distance gives a better representation of the closeness of two meshes than most of other distances, it is often hard to compute and can not be differentiated because it implies finding maxima of minima.

To avoid these issues, we use a distance field built such that the target mesh is its zero level-set [JBS06]. Distance fields have the advantages of being easy and fast to compute. They are differentiable everywhere and can be used to measure the distance between meshes of different topologies. As we compute the distance field corresponding to the target mesh, it can be pre-computed before tracking as, contrary to the model mesh, the target mesh does not evolve. Figure 4.4 shows an example of slices of a distance field for a visual hull from sequence ratC6X1M1lo.

Once the distance field is computed, we can define the distance function that has to be minimized. For a given mesh $\mathcal{M}$, the distance to the target $\mathcal{T}$ is computed as the sum of the distances of each vertex of $\mathcal{M}$ to $\mathcal{T}$. The distance of a vertex to $\mathcal{T}$ is the value of the distance field at the voxel to which the vertex belongs. Thus the distance

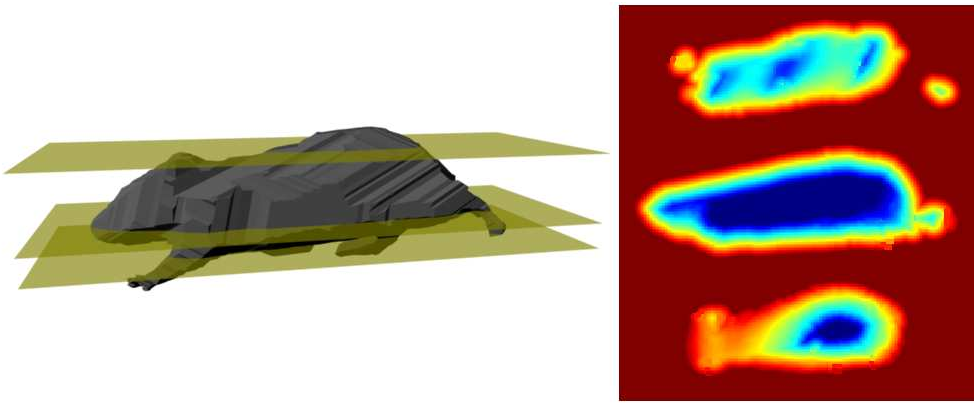


Figure 4.4: *Horizontal slices of a distance field (right) for a visual hull from sequence ratC6X1M1lo (left).*

function takes the following form :

$$d_{DF}(\mathcal{M}, \mathcal{T}) = \sum_{p_i \in \mathcal{M}} d_{DF}(p_i, \mathcal{T}) \tag{4.8}$$

where $d_{DF}(p_i, \mathcal{T})$ is the value of the distance field generated from $\mathcal{T}$ at position p_i.

4.3.2 Shape and pose tracking

To recover the deformation of the model, we could minimize the distance between the model and the target in the unconstrained highly-dimensional OQR space. To limit the evolution of the model to plausible deformations, we differentiate this distance in the latent space. As a result, we stay in the low-dimensional learned manifold i.e. in the space of probable OQR configurations.

We define an energy function to minimize with two terms. The first one measures how close the model and the target are. This is given by their distance as defined in Equation (4.8). The second term takes advantage of the fact that at each point in the latent space, we can compute the variance, which indicates how likely this latent point is to generate a valid data point. We use this variance as a regularization term to ensure that the shape generated from the latent point is as likely to exist as possible. We therefore try to find :

$$\hat{x} = \arg\min_{x}(d_{DF}(\mathcal{M}(x), \mathcal{T}) - \log(\sigma^2(x))) \tag{4.9}$$

where x is the latent point, $\mathcal{M}(x)$ is the model generated from the latent point x and σ^2 is the variance at point x.

Shape tracking

To find $\hat{x}$, we differentiate the energy of Equation (4.9) with respect to the latent variables. This is done by using the chain rule :

$$\frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{dx} = \frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{d\mathcal{M}}\frac{d\mathcal{M}}{d\mathcal{V}}\frac{d\mathcal{V}}{dy}\frac{dy}{dx} \tag{4.10}$$

where $\mathcal{V} = (v_0 \dots v_m)$ is defined in Section 3.3.1 as the vertices of the cage and $y \in \mathbb{R}^d$ is the OQR configuration generated from the latent point $x \in \mathbb{R}^q$ i.e. the set of positions and orientations of the r oriented quads.

$\frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{d\mathcal{M}}$ is computed by finite differences and can be pre-computed for each voxel.

$\frac{d\mathcal{M}}{d\mathcal{V}} = (\frac{dp_i}{dv_j})_{i=1..|\mathcal{M}|,j=|\mathcal{V}|}$ is computed by differentiating Equation (3.6) with respect to the vertices of the cage.

$$\frac{dp_i}{dv_j} = \Phi_{i,j}I_3 + \sum_{k,t_k=(j,k_0,k_1)} \Psi_{i,k} \begin{pmatrix} 0 & z_{k_1}-z_{k_0} & -(y_{k_1}-y_{k_0}) \\ -(z_{k_1}-z_{k_0}) & 0 & x_{k_1}-x_{k_0} \\ y_{k_1}-y_{k_0} & -(x_{k_1}-x_{k_0}) & 0 \end{pmatrix} \tag{4.11}$$

Each vertex v_i of $\mathcal{V}$ in V_c (see Section 3.3.2) is connected rigidly to one and only one oriented quad q_{j_i}. Therefore, if $\mathcal{V} = V_c$:

$$\frac{d\mathcal{V}}{dy} = (\delta_{j,j_i}\frac{dv_i}{dq_j})_{i=1..|\mathcal{V}|,j=1..r} \tag{4.12}$$

For vertices not in V_c, they are a linear combination of vertices from V_c (Equation (3.15)), their gradient is therefore the same linear combination of the gradients of the vertices in V_c. Following Equation (3.13), if the position of the quad q_{j_i} is T and its rotation is expressed by an axis/angle (e_x, e_y, e_z, α) with $\|e\| = 1$:

$$\begin{gathered} \frac{dv_i}{dT} = I_3 \\ \frac{dv_i}{d\alpha} = -sin(\alpha)v_i^0 + cos(\alpha)(e \times v_i^0) + sin(\alpha)(e.v_i^0)e \\ \frac{dv_i}{de} = sin(\alpha)\begin{pmatrix} 0 & Z_i^0 & -Y_i^0 \\ -Z_i^0 & 0 & X_i^0 \\ Y_i^0 & -X_i^0 & 0 \end{pmatrix} + (1-cos(\alpha))(e.v_i^0)I_3 + (1-cos(\alpha))ev_i^{0T} \end{gathered} \tag{4.13}$$

where $v_i^0 = (X_i^0, Y_i^0, Z_i^0)$ is a canonical vertex of Equation (3.11).

y is the mean of the Gaussian distribution by which the latent point x projects to a data point as expressed in Equation (4.3). Differentiating Equation (4.3) with respect to x gives :

$$\frac{dy}{dx} = Y^T K^{-1}\frac{dK_x}{dx} \tag{4.14}$$

Each element of K_x is the covariance function k applied on x and one learned latent point. As a result, $\frac{dK_x}{dx} = (\frac{dk(x,x_i)}{dx})_{i=1..n}$ follows from Equation (4.2) :

$$\frac{dk(x,x_i)}{dx} = -\gamma(x - x_i)k(x,x_i) \tag{4.15}$$

As for the regularization term, x projects to a data point with a mean as expressed in Equation (4.3) and a variance σ^2 as expressed in Equation (4.4). As a result :

$$\frac{d\sigma^2}{dx} = -2K_x^T K^{-1} \frac{dK_x}{dx} \tag{4.16}$$

Thanks to Equations (4.10) and (4.16), a simple first order method such as gradient descent can be used to minimize Equation (4.9) and recover the OQR configuration corresponding to the target.

Pose tracking

So far, we have captured the deformation of the shape as encoded in the latent space. As explained in Section 4.2, this shape is defined up to the global transformation Γ of a reference quad (Q2 in Figure 3.31 for the latent spaces of Figure 4.2 and Figure 4.3). As a result, we also need to recover the global translation and orientation of the model.

To do so, we minimize the energy without the regularization term by differentiating Equation (4.8) with respect to the pose parameters :

$$\frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{d\Gamma} = \frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{d\mathcal{M}} \frac{d\mathcal{M}}{d\Gamma} \tag{4.17}$$

where

$$\frac{d\mathcal{M}}{d\Gamma} = \begin{pmatrix} \frac{dp_1}{dT_x} & \frac{dp_1}{dT_y} & \frac{dp_1}{dT_y} & \frac{dp_1}{de_x} & \frac{dp_1}{de_y} & \frac{dp_1}{de_z} & \frac{dp_1}{d\alpha} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{dp_{|\mathcal{M}|}}{dT_x} & \frac{dp_{|\mathcal{M}|}}{dT_y} & \frac{dp_{|\mathcal{M}|}}{dT_y} & \frac{dp_{|\mathcal{M}|}}{de_x} & \frac{dp_{|\mathcal{M}|}}{de_y} & \frac{dp_{|\mathcal{M}|}}{de_z} & \frac{dp_{|\mathcal{M}|}}{d\alpha} \end{pmatrix} \tag{4.18}$$

if Γ is parameterised by a translation T and an axis/angle rotation (e_x, e_y, e_z, α) where $\|e\| = 1$ with :

$$\begin{aligned} \frac{dp_i}{dT} &= I_3 \\ \frac{dp_i}{d\alpha} &= -sin(\alpha)p_i^0 + cos(\alpha)(e \times p_i^0) + sin(\alpha)(e.p_i^0)e \\ \frac{dp_i}{de} &= sin(\alpha) \begin{pmatrix} 0 & Z_i^0 & -Y_i^0 \\ -Z_i^0 & 0 & X_i^0 \\ Y_i^0 & -X_i^0 & 0 \end{pmatrix} + (1 - cos(\alpha))(e.p_i^0)I_3 + (1 - cos(\alpha))ep_i^{0T} \end{aligned} \tag{4.19}$$

with $p_i^0 = (X_i^0 Y_i^0 Z_i^0)$ is the position of p_i in the rest pose.

Pose and shape optimization can be done either jointly or iteratively until convergence. Joint optimization is more correct but it converges more slowly as the search space is larger. Furthermore, experimentally, iterating between pose and shape optimization works faster and better, especially when tracking a sequence. In this case, the global transformation and the deformation between two frames are small enough that optimizing first for the pose does not lead to a pose from which finding the good deformation is impossible, and vice-versa.

Local adjustment of the OQR

One limitation of our method is its dependence on the learning data. As with every manifold-based method, we are limited to what the manifold can reconstruct. To account for this limitation, we can first optimize with respect to latent variables to get as close as possible to the target mesh. Once this has converged, a few more iterations can be done with respect to the oriented quads to locally adjust the cage by differentiating the energy function with respect to the oriented quads :

$$\frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{dy} = \frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{d\mathcal{M}} \frac{d\mathcal{M}}{d\mathcal{V}} \frac{d\mathcal{V}}{dy} \tag{4.20}$$

All these derivatives have been expressed previously as Equation (4.20) is part of Equation (4.10). Section 4.3.3 shows examples where local adjustment improve the tracking.

4.3.3 Validation and results

To validate our method, we show results on our datasets as well as on standard computer vision datasets. Unfortunately, standard computer vision datasets for which ground truth is known only exist for humans.

Validation on standard datasets

As standard datasets only exist for humans, we built an OQR for humans made of 21 oriented quads (Figure 4.5). We tested our method on two test sets. We first tested our approach on a temporally coherent mesh of high quality from [VBMP08]. We then tackled more challenging data by tracking a sequence of noisy visual hulls extracted from the HumanEva II dataset [SBB10], without temporal coherence, learning the motion manifold automatically from the CMU Motion Capture Database.

Figure 4.5: *Oriented Quads Rigging for humans.*

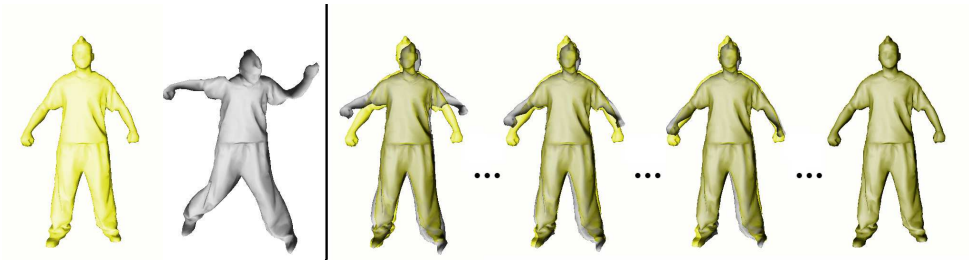


Figure 4.6: *(a) Target model - (b) Initial pose - (c) Samples on the convergence trajectory.*

We first tested our deformation optimization without the pose recovery. To do so, we use the bouncing sequence from [VBMP08]. The sequence of target meshes is made of a temporally coherent animated mesh of 10000 vertices and 20000 faces. We define the template mesh as one of these meshes simplified to 4000 vertices and 8000 faces. The training was done by manually locating the oriented quads on 16 frames of the sequence. We learn a 10-D manifold from these samples. Figure 4.6 shows the progression of the gradient descent towards an optimal deformation. We can notice that, since we are optimizing in the latent space, each of the sample deformations is a valid deformation.

Figure 4.7 shows the advantage of tracking in the manifold. As there is no structure on the cage, optimizing with respect to the oriented quads creates a shrinking of the cage as the quads are attracted to the closest patch of surface without any influence from the position and orientation of the other oriented quads. We can see on Figure 4.7 the influence of the regularization term of the energy. Without it, the model mesh is attracted to the shape that is the closest to the target mesh and can be generated from the latent space without any regard for the likelihood of this shape. Using the variance as a regularization term forces the optimal latent point to be more likely to generate a valid deformation.

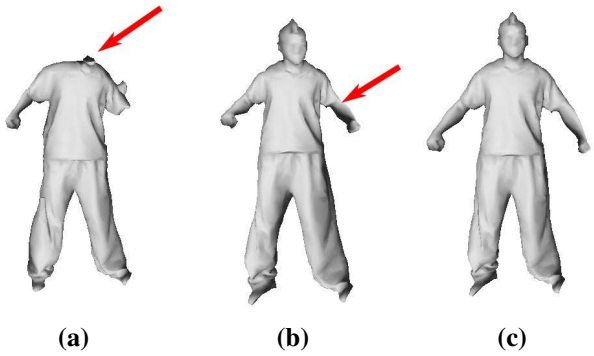


Figure 4.7: *With the same target model and initial pose as in Figure 4.6 - (a) Result of the optimisation in the oriented quad space. Notice how the head collapses - (b) Result of the optimisation in the 10-D latent space without any regularisation term. Notice how the arms shorten - (c) Result with the regularisation term.*

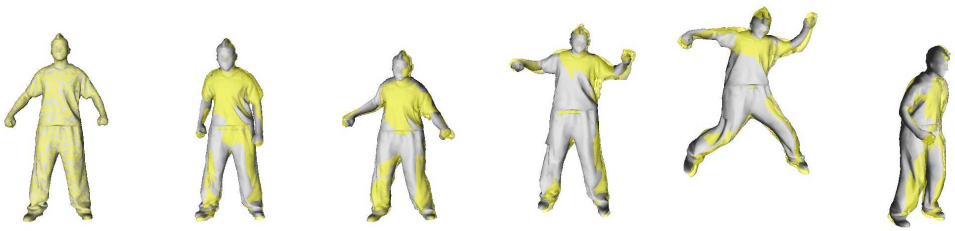


Figure 4.8: *Optimisation of both pose and deformation. The tracking (grey) is overlaid over the target (yellow).*

As explained in Section 4.3.2, we can also recover the global pose of the mesh. In practice, we noticed that pose recovery and deformation recovery done iteratively works better than joint optimization. Figure 4.8 shows results on the bouncing sequence.

We use the walk sequence of subject S4 in the HumanEva II dataset [SBB10] for a quantitative evaluation of our tracking as it contains the positions of joints acquired with markers and a Vicon system. This data consists of 4 synchronized color cameras. Using the background subtraction software and the mixture of Gaussian distribution model provided with the dataset, we extracted the silhouettes from the videos. From these silhouettes, we generated the visual hulls using the Exact Polyhedral Visual Hull method [FB03]. As the the camera locations are not optimal for a preprocess step using visual hull reconstruction, the visual hulls are a very coarse approximation of the observed shape as can be seen on the left of Figure 4.9. The model used for cage-based deformation is the mesh model for subject S4 generated by Stefano Corazza and the

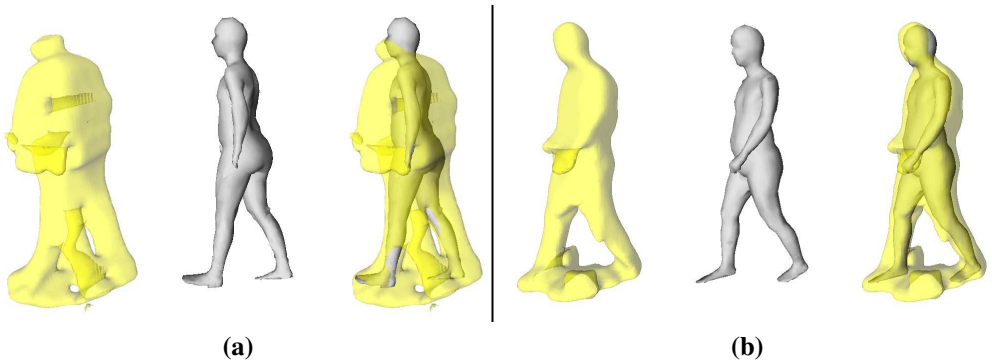


Figure 4.9: *Examples of tracking on the HumanEva II dataset : (from left to right) visual hull, result and overlay of the model on the visual hull. (a) The location of the cameras can lead the visual hull to be a very coarse approximation of the observed shape. (b) When the visual hull quality increases, the tracking becomes more precise.*

Stanford BioMotion Group using [CGMA10].

Instead of manually creating the learning data by manually fitting the OQR to meshes, we also propose a method to use an existing database of movements. From walking sequences in the CMU Motion Capture Database (http://mocap.cs.cmu.edu/), we automatically generate 120 training samples by rigidly mapping the quads of the OQR to the animation skeleton. It has to be done only once for a given frame of one of the subject's performance. After that, more samples can be automatically added to the training set using any available motion capture data samples for this subject. This training data is then used to build a 2D-manifold.

Figure 4.10 shows reprojections of the model tracked on the sequence. An example of local adjustment is shown in Figure 4.11, where a 2D latent space does not capture enough variance to track the mesh properly. Local adjustment allows for more precision once the mesh is already well fitted from the latent space.

Table 4.1 shows the 3D error as specified in the HumanEva II dataset with and without the local adjustment for the first 150 frames. This error is computed from the positions of the joints as computed from the markers. As our approach is not skeleton-based, the positions of the joints are computed by cage-based animation. On the first frame where motion capture data is available i.e. where the positions of the joints are known, the Green Coordinates are computed on these positions and then used to compute the new positions of the joints when the cage is deformed. With local adjustment, the mean error drops from 90mm to 38.6mm.

For the whole walking sequence, the error is 85.2mm (see Table 4.1). However, when considering only the first 150 frames, where the visual hulls are of better quality (right

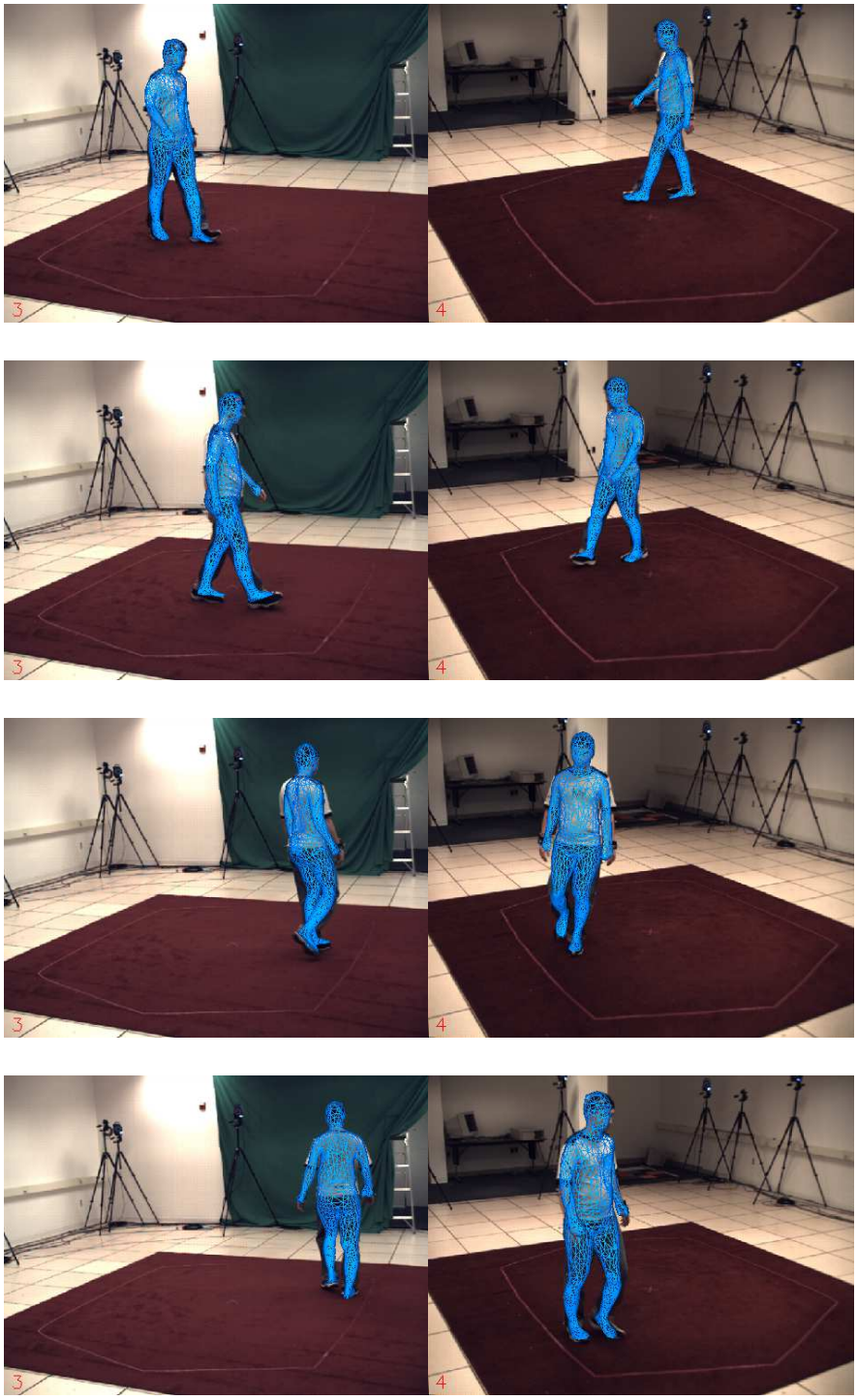


Figure 4.10: *Reprojection on camera 3 and 4 of the tracked model for frames 5, 45, 102 and 148 of the sequence for subject S4 in the HumanEva II dataset.*

of Figure 4.9), the error drops to 38.6mm. The error for the whole sequence is about the same as the error for the first 150 frames without local adjustment. This can be

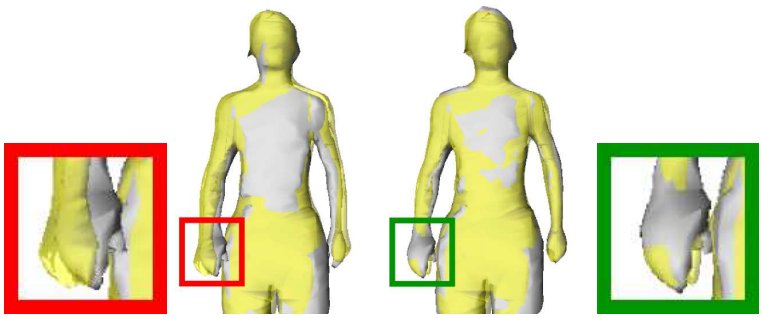


Figure 4.11: *Example of local adjustment of the oriented quads - (a) before local adjustment - (b) after local adjustment.*

Method	Frames	Mean error (mm)	Median error (mm)	Standard deviation (mm)
[CMG+10]	1-150	80.0		13.0
[GRBS10]	2-1258	32.01		4.53
Our method - no local adjustment	1-150	90.0	87.3	24.1
Our method	1-150	38.6	39.2	4.1
Our method	1-300	85.2	55.8	55.8

Table 4.1: *Error for sequence of subject S4 of the HumanEva II dataset. The mean, median and standard deviation in mm is measured for the first 150 frames without (third row) and with (fourth row) the local adjustment. The error increases after frame 150 (bottom row) as the quality of the visual hulls decreases (Figure 4.9) to reach the levels of tracking without local adjustment.*

explained by the non-discriminative quality of the target mesh. Indeed, the coarseness of the visual hull (left of Figure 4.9) prevents any fine adaptation of the shape by local adjustment. Table 4.1 also reports the results found in [CMG+10] and [GRBS10]. The method of Corazza et al. [CMG+10] also takes as input a series of visual hulls. The results are limited to the first 150 frames as re-initialisation is necessary after that due to the poor quality of the visual hulls. Thanks to the use of a manifold of deformations, even though the error increases after frame 150, we are still able to attain good results without re-initialisation. To our knowledge, Gall et al. [GRBS10] report the best results. However, they also use information not taken into consideration by our method such as the appearance of the model.

Results on locomotion tracking

We now focus on our rodent datasets, starting with sequence ratC6X1M1lo, which is locomotion of a rat. We therefore use the 2D latent space represented on Figure 4.2. As this sequence has 6 cameras in a good configuration for geometry reconstruction, the EPVH are of a better quality than the ones from the HumanEva II dataset. Figure 4.12 shows the result of tracking. The tracking is mostly correct, with the biggest imprecisions occurring for the foreleg. On the middle left image, the tracking is stuck in a local minimum as putting the leg further back would increase the error at the shoulder even though it would be a better fit for the paw. On the bottom left image, the EPVH is ambiguous on whether the foreleg is facing forward (as reconstructed) or backwards (as it should be). As both positions are possible and reachable in the latent space, the rest of the target does not help to disambiguate. On the bottom right image, we can see that the foreleg is not forward enough. This illustrates a flaw inherent to our method. Indeed, our energy function measures how close the model is to the target but does not take into consideration how close the target is to the model. As a result, this position of the foreleg gives an energy equivalent to what the correct position of the foreleg would give. We talk more about this issue in Section 4.5.

Figure 4.13 shows the results of Figure 4.12 reprojected on the top and side cameras of the set-up. We can see that due to the level of details of the OQR, the shape is not precisely matched (for instance the back of the animal). However, the motion is well tracked.

In the same way as we did for the HumanEva II dataset, we measure the quality of the tracking by comparing the positions of markers as tracked on the videos and as reconstructed from markerless tracking. In this case, Table 4.2 shows the mean and standard deviation of the distance between both for the external markers of Figure 2.22. To reconstruct the position of the markers from the OQR tracking, we compute the Green Coordinates of the landmarks in the rest pose and then proceed to update their positions by cage-based animation when the OQR configuration changes. To better understand this error, we also compute the same errors for body parts, only considering the markers mapped to a given body part. We can see that the error is quite high, slightly lower

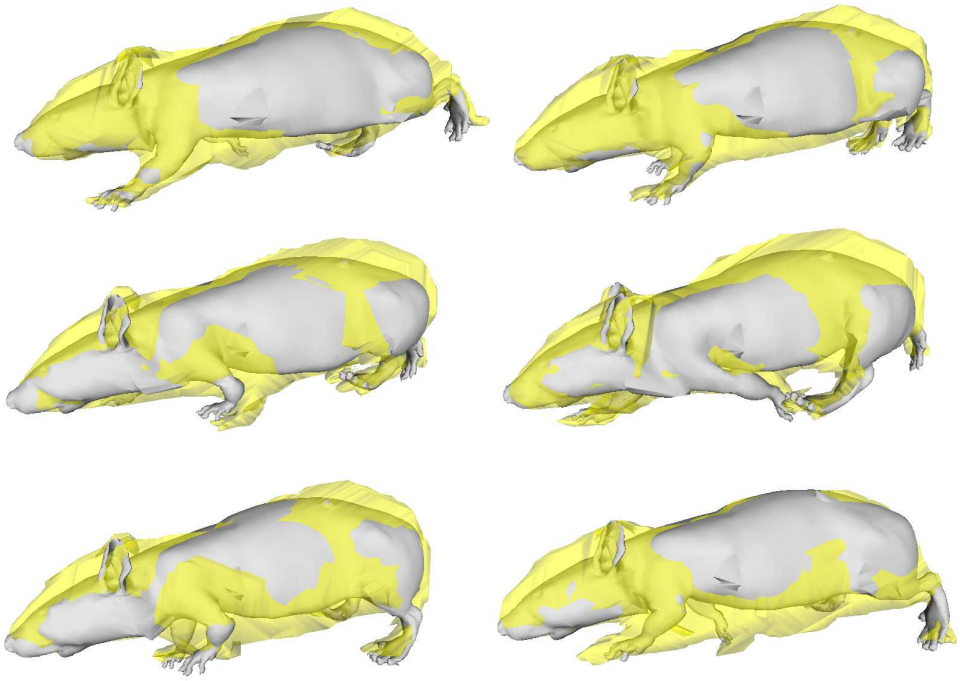


Figure 4.12: *Examples of tracking for sequence ratC6X1M1lo. The tracking (grey) is overlaid over the target (yellow).*

than 1cm. However, except for the foreleg, the standard deviation is very low, which leads to think that the error is mostly due to a bias. This bias probably comes from the fact that the model does not correspond perfectly to the studied animal or from a placement of the landmarks that is not perfect.

Markers :	All	Head (4 markers)	Foreleg (4 markers)	Hindleg (4 markers)
Mean error (mm) :	9.8	6.4	7.3	10.2
Standard deviation (mm) :	3.6	0.7	2.8	0.8

Table 4.2: *Mean and standard deviation of the 3D distance between the external markers of Figure 2.22 as tracked on the videos and as reconstructed from the tracked OQR for $n = 45$ frames.*

As explained in Section 3.4, once we have an animation of OQR, one can use the animation model of Chapter 2 to animate the bones from the skin. As simulated markers for weighted Inverse Kinematics, we use the manually chosen markers of Figure 3.33. Figure 4.14 shows the results for the frames seen on Figure 4.12 and Figure 4.13. Interestingly enough, we had labeled the third and fifth examples as incorrectly tracked for the foreleg. However, on Figure 4.14, we can see that, even though on the fifth

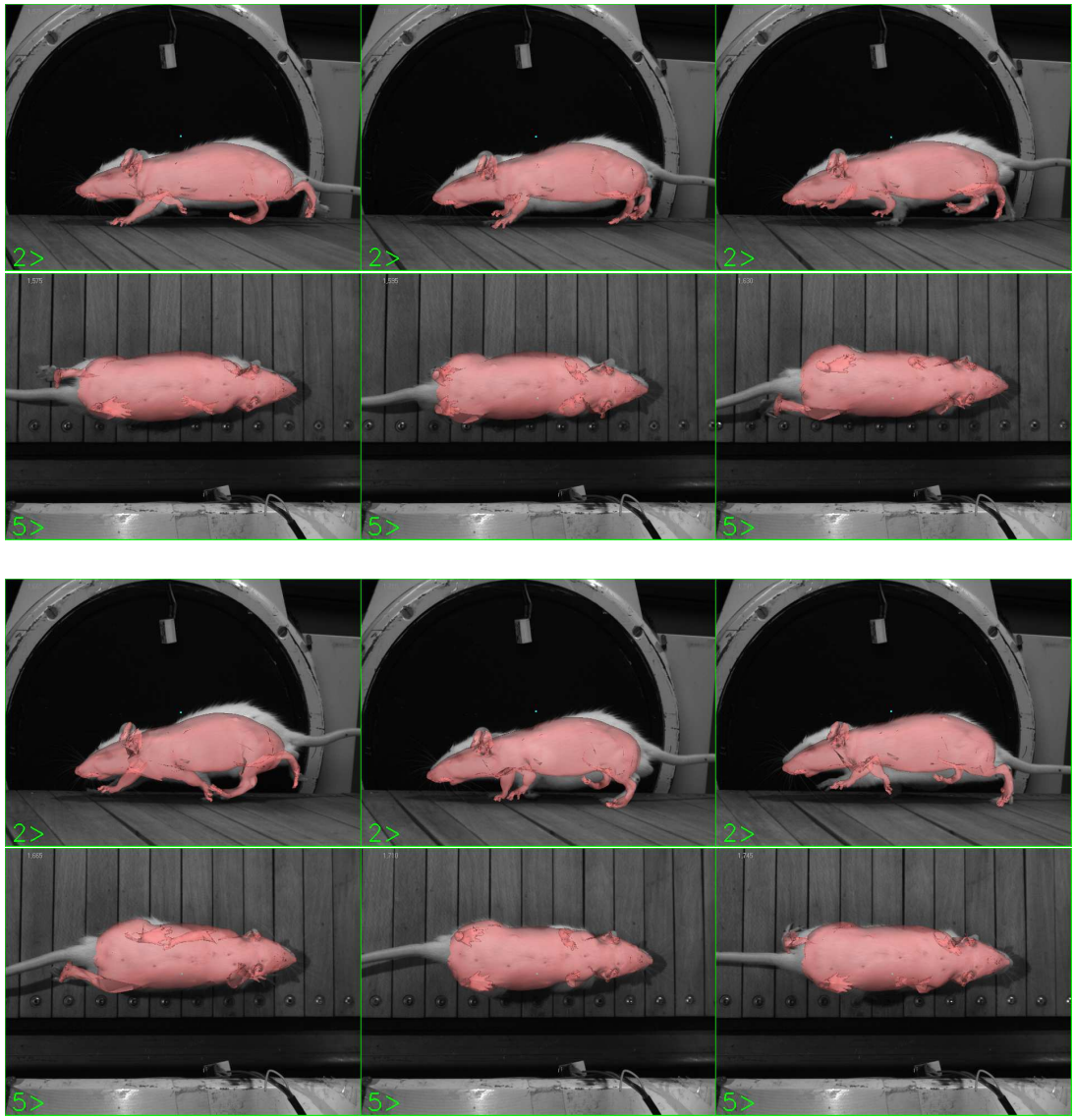


Figure 4.13: *Examples of Figure 4.12 reprojected on the side and top cameras.*

example, the foreleg remains incorrectly tracked, on the third example, the radius is positioned outside the skin but closer to the truth. This is due to the angular springs that impose anatomical constraints. As the shape of the envelope on the third example is only achievable by the animation model of the bones by large elongations of the angular springs, a solution closer to the rest pose but farther away from the markers is chosen. As explained previously, the error of the fifth example is due to an ambiguity of the visual hull but leads to a plausible result, which is why the anatomical constraints in the weighted Inverse Kinematics can not correct this error.

As in Section 2.5.2, the internal markers tracked in 2D on the X-ray video can be used to evaluate our tracking. As Figure 2.45 compared the projection of the landmarks to

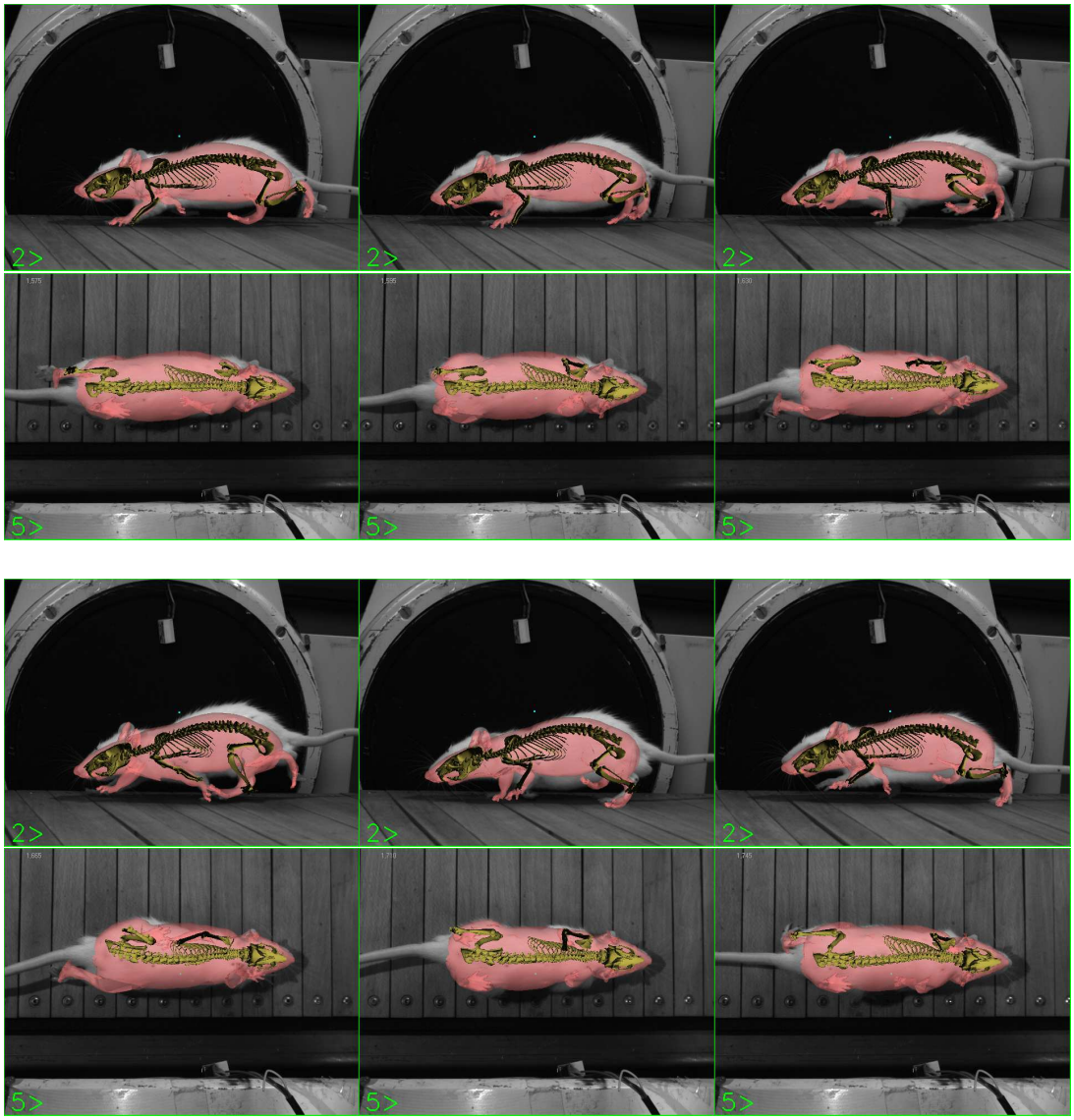


Figure 4.14: *Examples of Figure 4.12 reprojected on the side and top cameras with the result of weighted Inverse Kinematics from the skin as explained in Section 3.4.*

the tracking of the internal markers for marker-based tracking, Figure 4.15 compares the projection of the landmarks on the X-ray video to the 2D tracking of the internal markers for markerless tracking. Comparing both, we can see that, even though the bias is still here, the results are less precise with markerless tracking. Quantitatively, whereas we had a mean error of about 15 pixels, which is equivalent to 2.4mm, with marker-based tracking, we have an error of 49 pixels, which is equivalent to 7mm, with markerless tracking. However, given the fact that the error of the tracking of the skin was measured to be 9mm, it seems as if using the vertices of the skin for weighted Inverse Kinematics does not lead to further errors than the ones due to skin tracking. On the contrary, as shown in Figure 4.14, our Inverse Kinematics method sometimes

correct some of these errors because of anatomical constraints.

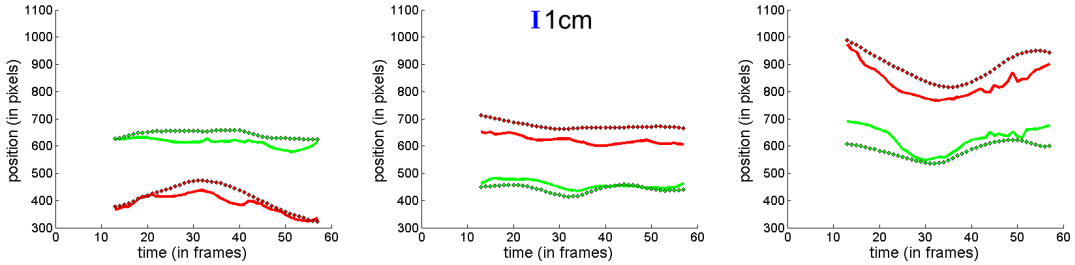


Figure 4.15: *Horizontal (red) and vertical (green) coordinates of the reprojection on the X-ray of the landmarks as tracked with our markerless technique (line) compared to the tracking of the corresponding internal markers (dots) on the X-ray video for sequence ratC6X1M1lo. From left to right - marker 13 (humerus), 16 (spine) and 24 (knee). See Figure 2.22 for the exact location of the markers.*

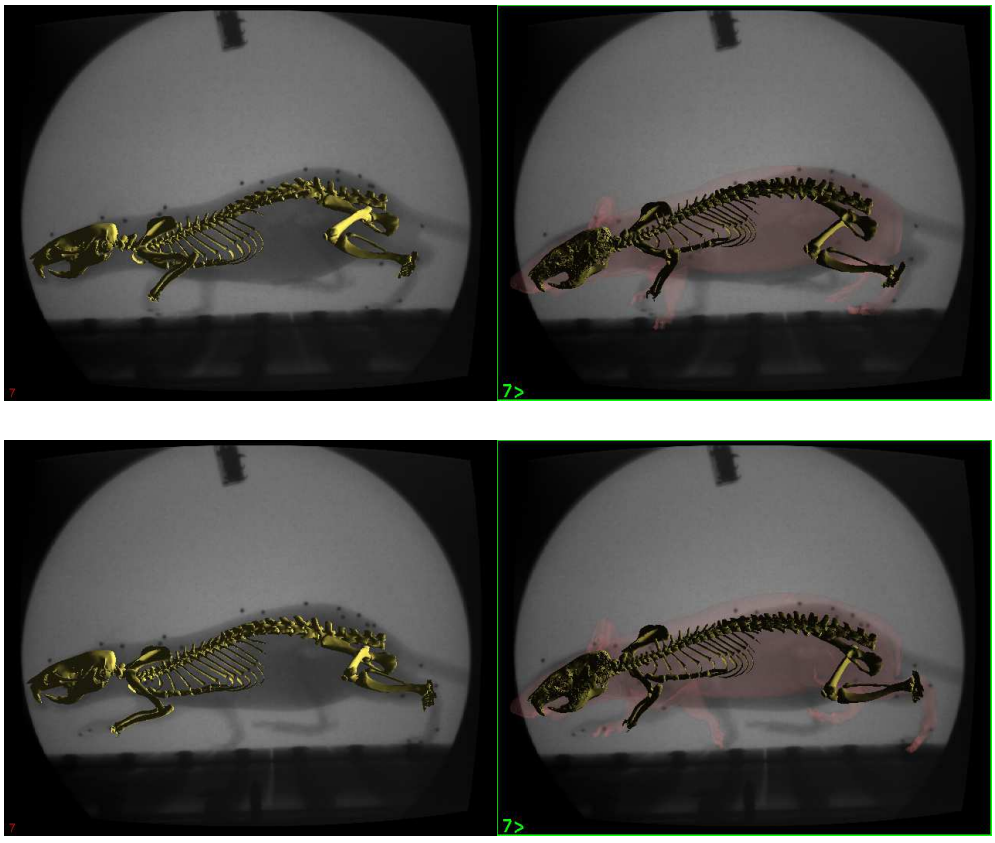


Figure 4.16: *Comparison between marker-based (left) and markerless (right) tracking of the internal structures.*

Figure 4.16 shows side by side projections on the X-ray video of marker-based and markerless tracking. The top row shows a frame where both are very similar, even though markerless tracking is a bit less precise. The bottom row shows an example where the humerus and radius are badly tracked in markerless tracking because the skin is not correctly fitted and where the knee does not elevated enough because, for markerless tracking, the stiffness for the angular springs is chosen higher than for marker-based tracking to compensate for skin tracking errors.

Anatomical entity	Tx	Ty	Tz	Rx	Ry	Rz
head	0.7885	0.3984	0.3014	0.4370	0.3703	0.3919
spine1	0.7241	0.7672	0.5932	0.7356	0.7494	0.8314
spine2	0.5518	0.4406	0.3259	0.5394	0.2831	0.6076
spine3	0.4441	0.6399	0.7975	0.60337	0.5652	0.5750
spine4	0.7731	0.3465	0.8401	0.6754	0.2073 (a)	0.2527
scapula	**0.9216**	0.6774	**0.9665**	**0.9592** (e)	0.2446 (b)	0.5950
humerus	0.8448	0.8169	**0.9575**	**0.9797** (f)	0.7817	0.5241
radius-ulna	0.2240 (c)	0.7466	0.7223	0.8737	0.7239	0.6922
femur	0.8218	0.1834	**0.9754** (g)	**0.9459**	0.4538	**0.0893** (d)
tibia	**0.9246**	0.4576	**0.9719** (h)	**0.9535**	0.3342	0.5098

Table 4.3: *Correlation coefficients of the degrees of freedom of the solids. Z is up, X is the direction along which the rat is walking and Y is perpendicular to the gait path (see Figure 1.8) for* $n = 45$ *frames.*

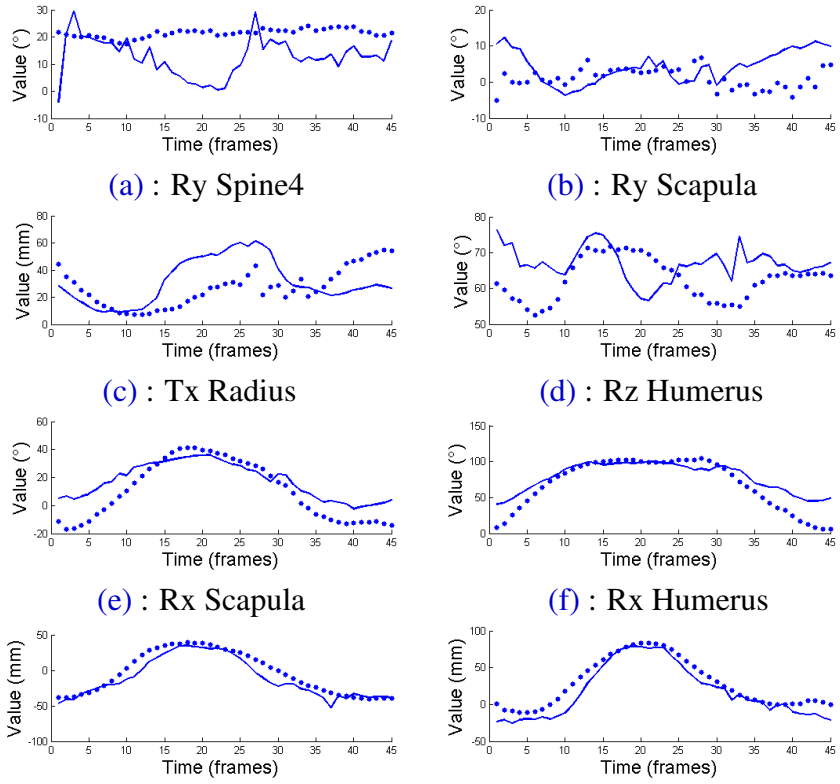
To quantify the difference between marker-based and markerless tracking of internal structures, we compare the positions and orientations of the solids in world space. Table 4.3 shows the correlation coefficients for the degrees of freedom of the different entities between marker-based and markerless tracking. Z is up, X is the axis along which the rat walks and Y is perpendicular to the gait path (see Figure 1.8). In red are the degrees of freedom for which the correlation is not significant. This is computed by computing the p-value for testing the hypothesis of no correlation. This p-value is the probability that we would have found the current result if the true correlation coefficient was 0. If the p-value is smaller than 5%, we consider the correlation to be significant. In our case, the p-value is larger than 5% for :

- (a) : rotation about Y for two solids of the spine. This problem is largely corrected by the use of the Hermite spline as the other solids are correctly tracked ;
- (b) : rotation about Y of the scapula. It can be seen on the Figure 4.16 that as the scapula is reconstructed slightly too high, it does not have to rotate as much around Y to fit the humerus and radius properly. Adding more vertices as markers in the weighted IK may solve the problem ;
- (c) : translation on X of the radius, which is something we noticed before with the fact that the radius does not go forward enough. This could be corrected by

tracking the wrist, which is characteristically at the border of the fur and skin, on the images and adding its position as a tracking constraint ;

- (d) : translation on Y and rotation about Z of the femur, which we noticed on Figure 4.16 as the higher stiffness in rotation of the angular springs does not allow the femur to properly lift. Decreasing the stiffness of that particular articulation may improve the tracking.

Figure 4.17 shows the degrees of freedom of the solids where the correlation is very low (first two rows) or very high (last two rows) for both methods on 1 cycle of locomotion. Even though, the degrees of freedom are very close, the animation curves for markerless tracking (lines) seem flatter than those for marker-based tracking (dots). This is in accordance with our previous observation that tracking is less precise because of the need to use a higher stiffness in rotation for angular springs. Indeed, that leads to minute movements not being tracked.



(a) : Ry Spine4 (b) : Ry Scapula

(c) : Tx Radius (d) : Rz Humerus

(e) : Rx Scapula (f) : Rx Humerus

(g) : Tz Femur (h) : Tz Tibia

Figure 4.17: *Comparison of the degrees of freedom of the solids as reconstructed with our marker-based (dots) and markerless (line) tracking. Z is up, X is the direction along which the rat is walking and Y is perpendicular to the gait path (see Figure 1.8).*

Results on open-field tracking

A challenging and interesting dataset is moC8X0M0of as the mouse can roam freely on a flat square. A 12-D latent space is built from 12 samples. Figure 4.18 shows some examples of results of mesh tracking on this sequence and Figure 4.19 shows the corresponding reprojection on 2 of the 8 cameras. As the mouse only occupies a small percentage of the 640×480 images (see Figure 1.10), the visual hulls are not very precise. As a result, even though the body and the head are correctly registered, the legs are often not properly fit unless they really stand out of the body.

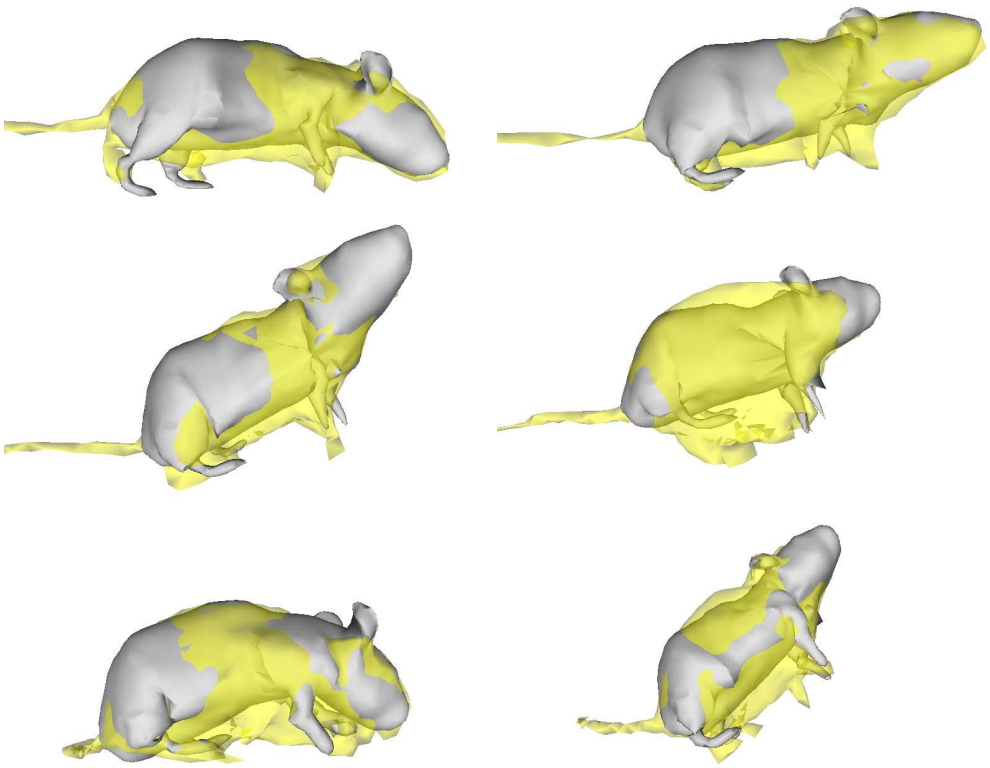


Figure 4.18: *Examples of tracking for sequence moC8X0M0of. The tracking (grey) is overlaid over the target (yellow).*

Figure 4.20 shows the result of weighted Inverse Kinematics reprojected on a third camera. A set of vertices equivalent to that of Figure 3.33 is used as markers.

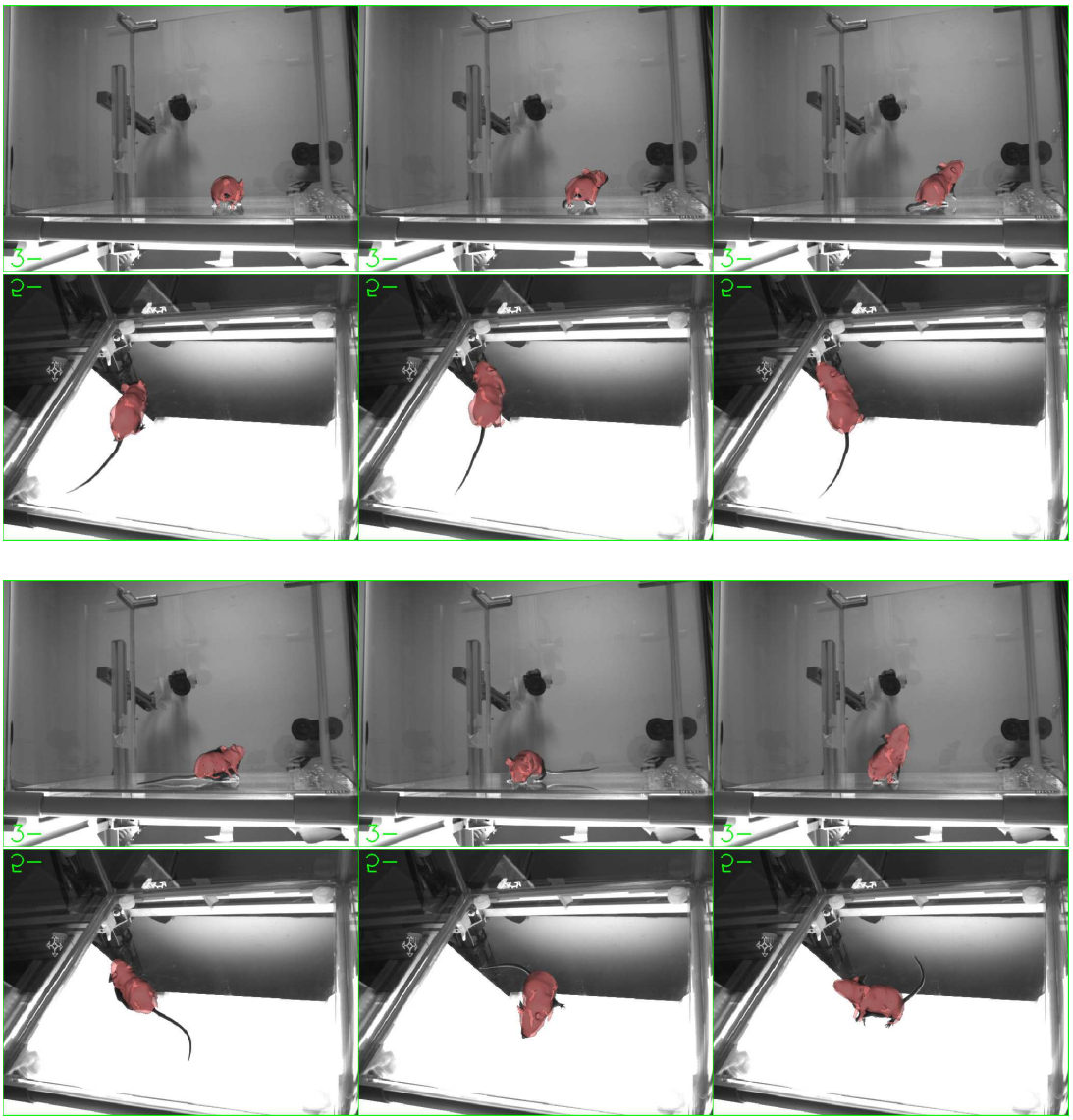


Figure 4.19: *Examples of Figure 4.18 reprojected on two cameras.*

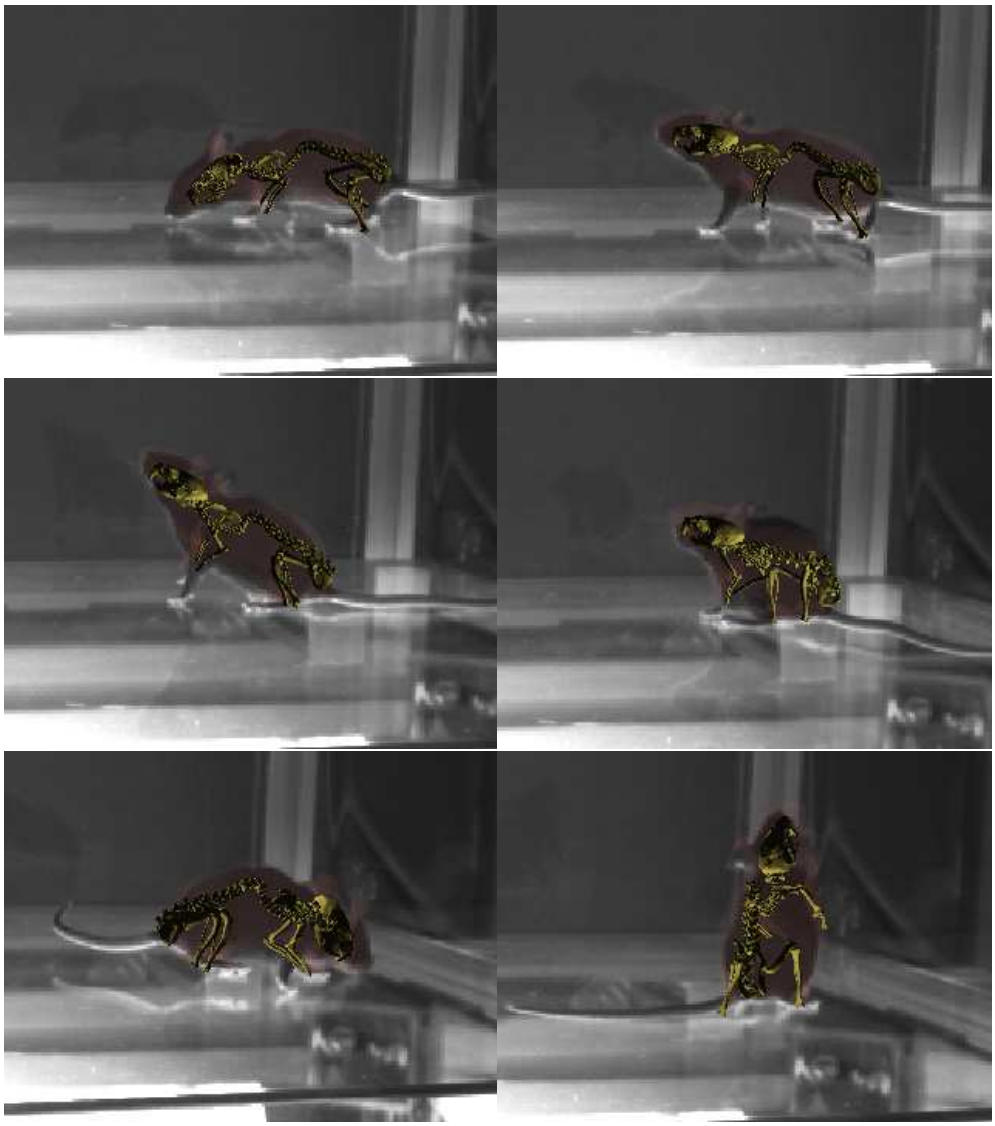


Figure 4.20: *Examples of Figure 4.19 with weighted Inverse Kinematics reprojected on another camera.*

4.4 VIDEO TRACKING

As has been shown in Section 3.2, if there is a false-negative in one silhouette or if the reconstructed object is partially out of one image then the corresponding geometry is not reconstructed. As a result, when dealing directly with meshes, we lose some information that was in the images. For instance, if for one camera, the head of the animal is out of the viewing volume whereas it is visible in all other cameras, it will not be reconstructed in the EPVH and the mesh will have no information on the head. Even though manifold learning helps tackling these issues as seen with the HumanEva II dataset (Section 4.3.3), it motivates the idea of directly working on images so as to not lose any information. This is further motivated by the fact that adding other cues such as appearance has been demonstrated to improve tracking by works such as [GRBS10]. Working directly on images makes the use of appearance cues easier. Therefore, we adapt our markerless tracking of Section 4.3 to an energy function computed on the images, not the resulting visual hull.

4.4.1 2D objective function

To directly compute an energy function on the images, we need to have a measure of how well the 3D mesh of the model fits what is observed on the images. For this, given a camera c, the error of the model $\mathcal{M}$ is measured on this cameras as :

$$d_{DF,c}(\mathcal{M}) = \sum_{p_i \in \mathcal{M}} d_{DF,c}(proj_c(p_i)) \tag{4.21}$$

where $proj_c(p_i)$ is the projection of 3D point p_i on camera c and $d_{DF,c}((u\ v))$ is the distance at pixel (u, v) to the silhouette on camera c. As for the 3D distance field, the 2D distance field can be pre-processed. Figure 4.21 shows examples of this distance.

For one set-up with n_c cameras, the overall error is computed as :

$$d_{DF}(\mathcal{M}) = \sum_{c=1..n_c} \delta_v d_{DF,c}(\mathcal{M}) \tag{4.22}$$

where δ_v is a visibility flag. This visibility flag is set to 1 if more than 75% of the vertices of $\mathcal{M}$ projects back within the image, 0 otherwise. In the following, we note $\mathcal{Z} = (proj_j(p_i))_{p_i \in \mathcal{M}, j=1..n_c}$.

4.4.2 Modifications to 3D shape and pose tracking

As in Section 4.3.2, to recover the deformation of the model, we minimise Equation (4.22) by differentiating it with respect to the latent space. Applying chain rules gives us :

$$\frac{d(d_{DF}(\mathcal{M}))}{dx} = \frac{d(d_{DF}(\mathcal{M}))}{d\mathcal{Z}} \frac{d\mathcal{Z}}{d\mathcal{M}} \frac{d\mathcal{M}}{d\mathcal{V}} \frac{d\mathcal{V}}{dy} \frac{dy}{dx} \tag{4.23}$$

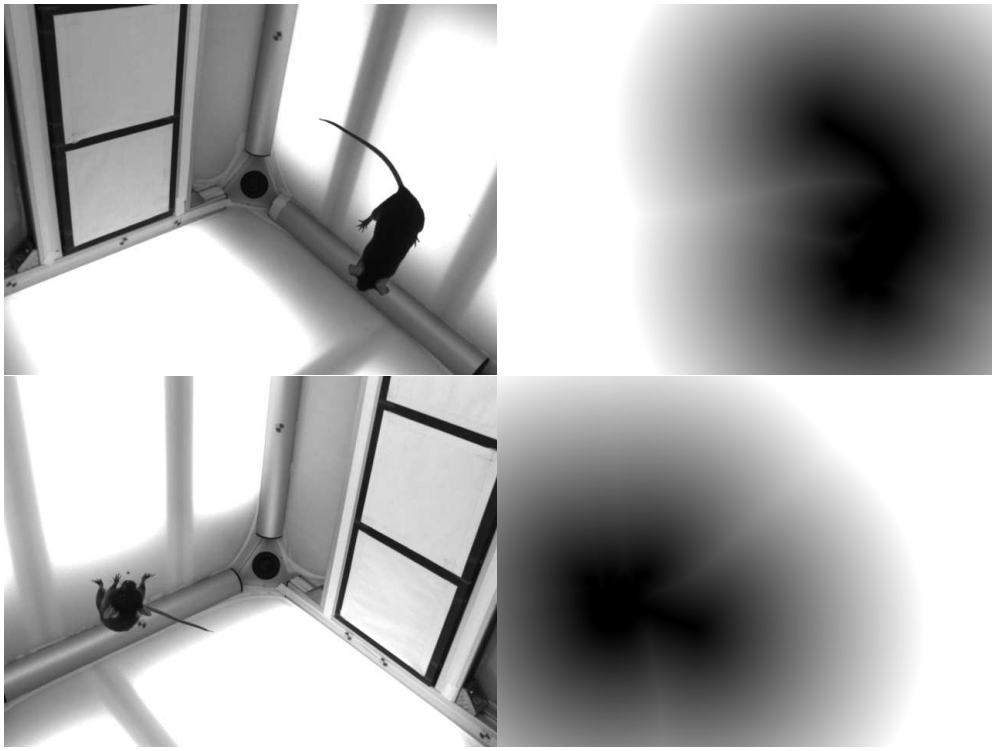


Figure 4.21: *Examples for two images (left) of the distance to the silhouette : the darker the pixel, the closest to the silhouette the pixel is (right).*

for shape tracking,

$$\frac{d(d_{DF}(\mathcal{M}))}{d\Gamma} = \frac{d(d_{DF}(\mathcal{M}))}{d\mathcal{Z}} \frac{d\mathcal{Z}}{d\mathcal{M}} \frac{d\mathcal{M}}{d\Gamma} \tag{4.24}$$

for pose tracking and :

$$\frac{d(d_{DF}(\mathcal{M}))}{dy} = \frac{d(d_{DF}(\mathcal{M}))}{d\mathcal{Z}} \frac{d\mathcal{Z}}{d\mathcal{M}} \frac{d\mathcal{M}}{d\mathcal{V}} \frac{d\mathcal{V}}{dy} \tag{4.25}$$

for local adjustment. We see that the only difference with Equations (4.10), (4.17) and (4.20) is that $\frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{d\mathcal{M}}$ is replaced by $\frac{d(d_{DF}(\mathcal{M}))}{d\mathcal{Z}} \frac{d\mathcal{Z}}{d\mathcal{M}}$ in all three equations.

$\frac{d(d_{DF}(\mathcal{M}))}{d\mathcal{Z}}$, just like $\frac{d(d_{DF}(\mathcal{M},\mathcal{T}))}{d\mathcal{M}}$, is computed by finite differences that can be be pre-processed.

$\frac{d\mathcal{Z}}{d\mathcal{M}}$ is the gradient of the projections onto the different cameras of the points of $\mathcal{M}$. As such it can be written as :

$$\frac{d\mathcal{Z}}{d\mathcal{M}} = (\delta_{i,j} \frac{d(proj_c(p_j))}{dp_i})_{i=1..|\mathcal{M}|,j=1..|\mathcal{Z}|,c=1..n_c} \tag{4.26}$$

Every point (X_c, Y_c, Z_c) in the camera canonical reference frame projects to a 2D point :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f\frac{X_c}{Z_c} + u_0 \\ f\frac{Y_c}{Z_c} + v_0 \end{pmatrix} \tag{4.27}$$

where f is the focal length of the camera and (u_0, v_0) is the principle point on the image plane. For a point (X, Y, Z) in the world reference frame, its coordinates in the camera reference frames are computed as :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + T \tag{4.28}$$

where (R, T) are the rotation and translation of the camera. This gives us :

$$\frac{d(proj_c(p_i))}{dp_i} = \begin{pmatrix} f\frac{R_{0,0}Z_c - R_{2,0}X_c}{Z_c^2} & f\frac{R_{0,1}Z_c - R_{2,1}X_c}{Z_c^2} & f\frac{R_{0,2}Z_c - R_{2,2}X_c}{Z_c^2} \\ f\frac{R_{1,0}Z_c - R_{2,0}Y_c}{Z_c^2} & f\frac{R_{1,1}Z_c - R_{2,1}Y_c}{Z_c^2} & f\frac{R_{1,2}Z_c - R_{2,2}Y_c}{Z_c^2} \end{pmatrix} \tag{4.29}$$

4.4.3 Comparison with mesh tracking

We first compare the mesh tracking of Section 4.3 with our video tracking. In case of perfect silhouettes, the results are equivalent. Figure 4.22 shows the result on a frame of sequence ratC6X1M1lo for both methods initialised with the same pose and latent point. The results are equivalent. Indeed, in the 2D latent space where the training samples are within $[-3.17, 0.69] \times [-1.65, 1.75]$, from the initial latent point $(-1.93, 0)$, mesh tracking converges to the latent point $(-2.77, -0.15)$ and video tracking converges to $(-2.66, -0.26)$.

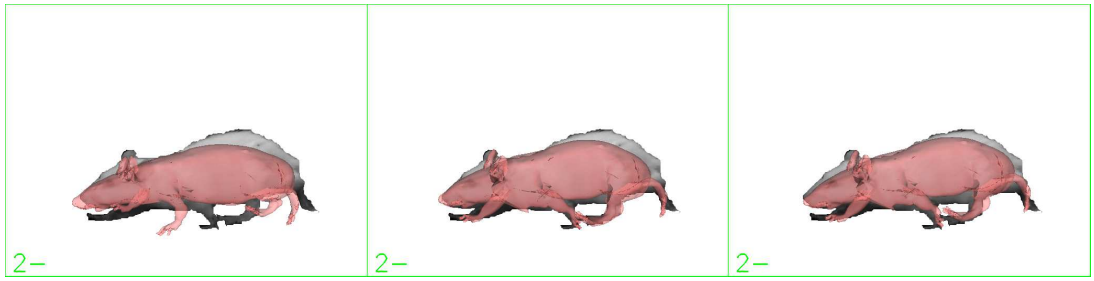


Figure 4.22: *From left to right : initialisation - result of mesh tracking - result of video tracking.*

However, if one of the silhouettes is noisy, video tracking is more robust than mesh tracking. For instance, on the same frame as Figure 4.22, we deteriorate the silhouette of one of the six cameras by removing the legs. The right column of Figure 4.23 shows

those silhouettes and the resulting visual hull reconstruction. As the visual hull reprojects exactly in the silhouettes of all cameras, the target mesh does not have any geometry for the missing legs. For video tracking, even though the legs have disappeared in that particular camera, the silhouettes of the other cameras still contain information about the missing limb. Figure 4.23 shows the results for both mesh and video tracking when removing 1 to 3 limbs on one camera. Table 4.4 shows the latent point to which the tracking converges to for the same data. Thanks to the fact that tracking is done in the learned manifold, both mesh and video tracking locate the missing limb correctly when only one leg is missing. However, when the two limbs of the left side of the rat are missing, mesh tracking begins to deteriorate. Indeed, the visual hull enables the tracking to correctly locate the right side of the animal but does not disambiguate between several shapes that can be created by the latent space. As for video tracking, as the information about the missing limbs is still present in the silhouettes from the other cameras, the tracking does not deteriorate. Video tracking is thus more robust than mesh tracking when dealing with false negatives in background subtraction.

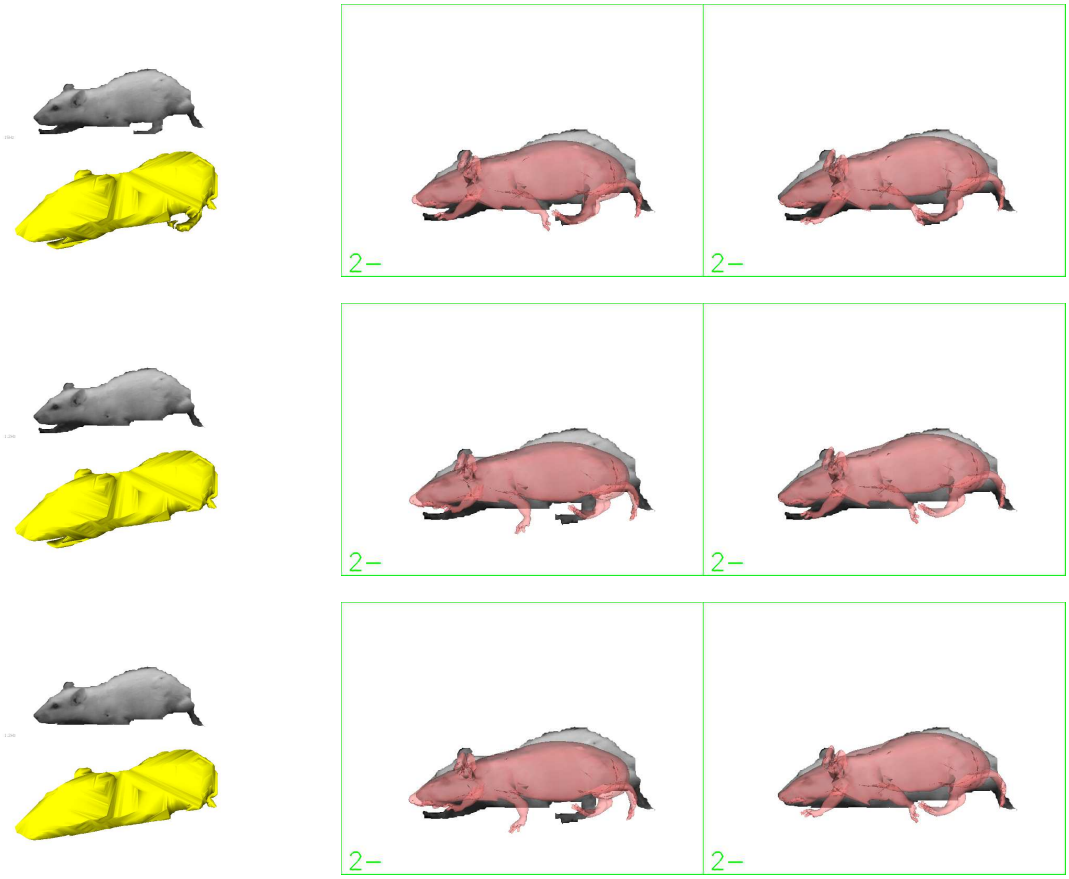


Figure 4.23: *From left to right : deteriorated silhouette and resulting EPVH - result of mesh tracking - result of video tracking.*

erased from 1 camera	mesh tracking result	video tracking result
0 leg	$(-2.77, -0.15)$	$(-2.66, -0.26)$
1 leg	$(-2.63, -0.17)$	$(-2.85, -0.15)$
2 legs	$(-2.23, -0.05)$	$(-2.66, -0.25)$
3 legs	$(-2.10, 0.12)$	$(-2.65, -0.14)$

Table 4.4: *Latent point to which mesh and video tracking converge when the silhouettes are deteriorated by removing the limb on the silhouette of 1 camera.*

4.4.4 Results on parabolic flights

As the silhouettes are noisy on the parabolic flight sequences moC9X0M0pf (see Figure 3.14) and therefore lead to visual hulls with missing information (see middle row of Figure 3.17), we use video tracking rather than mesh tracking. The latent space is learned from 8 manually labeled samples (Figure 4.3). Figure 4.24 shows two different frames. The top row shows an example where the tracking is a success. The bottom row shows an example where the tracking fails. This is due to several reasons. First of all, the latent space is learned from 8 samples selected over all frames of all parabolic flights. As a result, it does not capture all the shapes that are adopted by the mice. This could be corrected by labeling more samples.

However, there is a more fundamental issue that leads to failures. Our objective function is vertex-based. It projects every vertex of the model and each vertex contributes the same. Given the resolution of the images, body parts such as the feet that project in a few pixels have the same influence as part of the back. As a result, the error when the back is inside the silhouette but the feet not correctly located is the same or can even be lower than when the feet are correctly located but the back of the animal is slightly outside the silhouette because the deformation model does not properly fit the real mouse. Thus, the issue comes from the fact that details give the same order of error as imprecisions in the deformation model. A weighting scheme to favour some vertices could be considered to overcome this problem.

Consequently, the results are not good enough to be able to do measures on the shape. However, the results are good enough to hope to differentiate characteristic behaviours of certain strains of mice. For instance, one can observe that the control mice adopt a good orientation and a good posture i.e. shape of their body when going from 0g to 2g. They curve their back and spread their limbs to land on their feet (left of Figure 4.25). On the other hand, even though they adopt a good orientation, ied mice do not adopt a good posture for landing (middle of Figure 4.25). As for IsK mice, they do not prepare for landing and usually crash on the floor (right of Figure 4.25).

To see if we could automatically differentiate between strains, we develop a landing

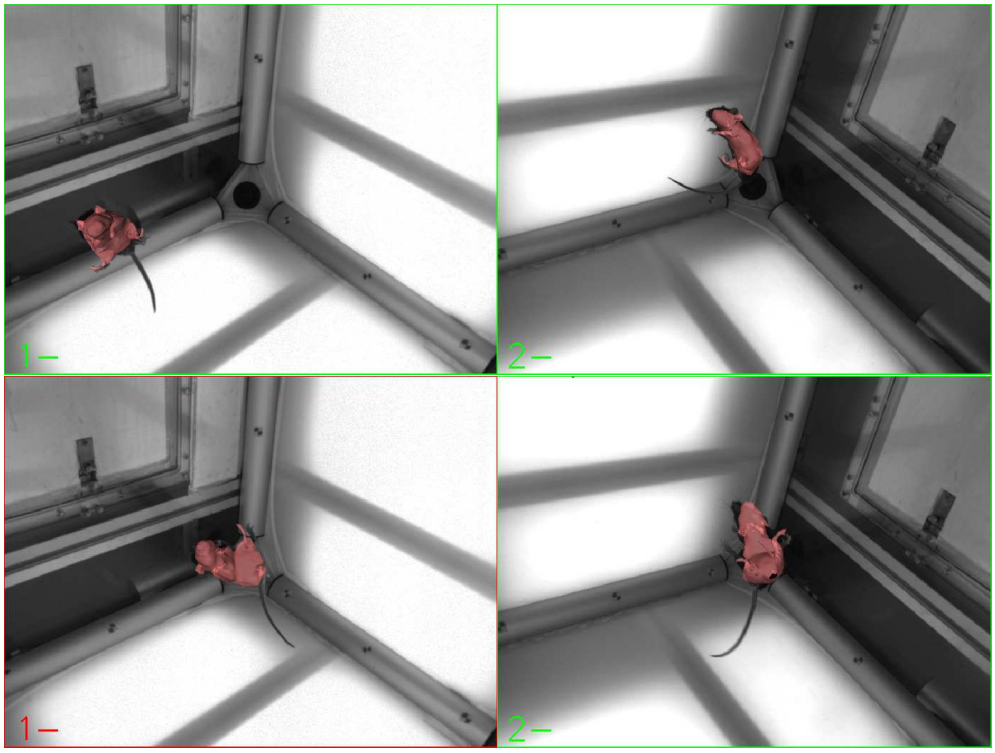


Figure 4.24: *Examples of success (top) and failure (bottom) of video tracking on moC8X0M0pf.*

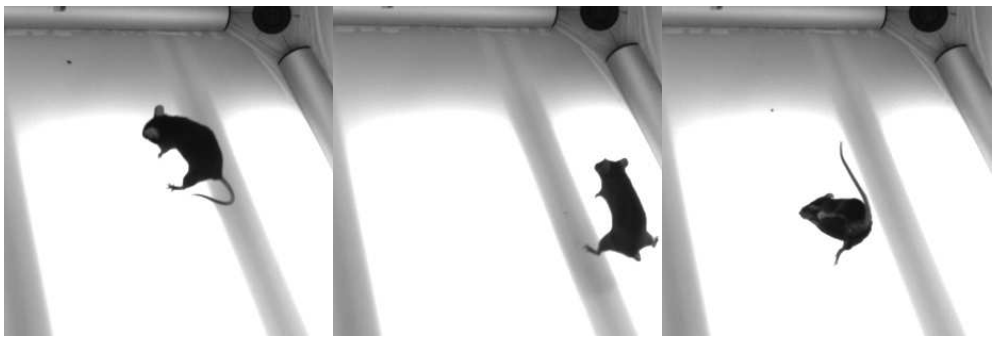


Figure 4.25: *Characteristic orientation and posture when going from 0g to 2g for control mice (left), ied mice (middle) and IsK mice (right).*

score, that is a measure of how well the mouse prepares for landing. The closer to 0 the score is, the more adapted for landing the orientation and posture of the mouse are. This score is computed as :

$$score_{landing} = score_{pose} * score_{spine} * score_{legs} \tag{4.30}$$

$score_{pose}$ measures how fitted the orientation is for landing by checking how parallel to the floor the mouse is. This is measured by the angle between the normal to the floor and the normal of the plane defined by the quads Q0 (nose), Q13 and Q16 (hindlegs). See Figure 3.31 for their location.

$score_{spine}$ measures the curvature of the spine. It is measured as the inverse of the angle between the axis (Q0,Q2), the thorax, and the axis (Q2,Q4), the abdomen.

$score_{legs}$ measures how well the legs are located for the impact. This is measured by projecting Q2 (the center of mass) on the axis (Q13,Q16), the axis between the two hind legs. Let's call $P2$ that projection. The measure is then computed as

$$score_{legs} = d(Q13, Q16)^{-1} \left| 1 - \frac{d(P2, Q13)}{d(P2, Q16)} \right| \tag{4.31}$$

It checks that the hindlegs are extended and symmetrical.

Figure 4.26 shows the landing score for 3 mice of the each strain. One can see that only control mice are able to reach and, above all, maintain a good orientation and posture prior to landing.

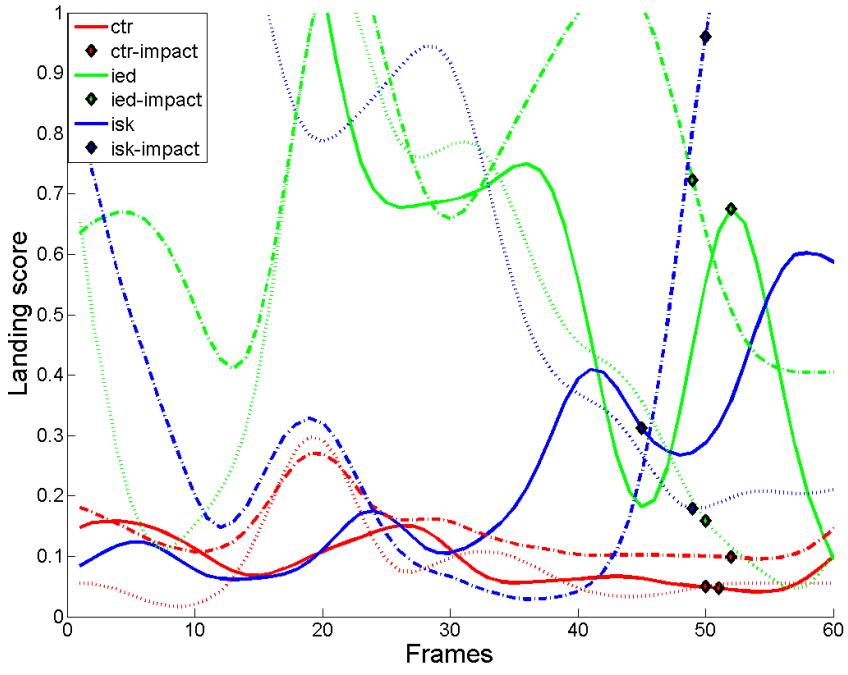


Figure 4.26: *Landing score for 3 mice of each strain. The points represent the moment of impact with the floor.*

4.5 CONCLUSION

From the deformation model proposed in Chapter 3, we have derived a mesh tracking algorithm that can work with both a sequence of meshes or segmented images as inputs. This tracking algorithm is regularised by a learned manifold of plausible OQR configurations, leading to a more robust solution. Experimental results on standard human datasets demonstrate the validity of our approach. The quality of rodent tracking highly depends on the resolution of the input data. However, given the same quality of input data, rodent tracking performs as well as human tracking.

One major advantage of using an OQR representation over using a skeleton representation is that the different morphologies of the animals can be encoded in the same manifold. Indeed, unlike joints in a skeleton that are defined locally in the hierarchy, oriented quads are defined globally. As a result, whereas morphology is encoded by different parameters in skeletons, morphology is encoded implicitly in the OQR parameters and as a result can be directly integrated in the latent space.

However, our current method has not considered the fact that classic GP-LVM struggles with learning a good model for complex datasets and becomes intractable for large datasets as one containing both motion and morphology information would be. Exploring other dimensional-reduction techniques or using a stochastic approach as in [YGvGU11] may overcome these issues and lead to a better tracking on complex sequences.

In terms of performance, the main reason for failure seems to be the fact that we are using a 'one-way' distance. Indeed, we measure how far the model is from the target but not how far the target is from the model. As a result, our tracking algorithm sometimes gets stuck in a local minimum where the surface of the model is close to the surface of the target but some parts of the target gets unaccounted for. This can happen both with the 3D or 2D objective function. For example, on the bottom right image of Figure 4.24, the right hindleg of the animal is within the silhouette. However, no part of the mouse projects back into the observed right hindleg. Using a 'two-way' distance would be better but would increase the computation time and make the gradient derivation harder. Some more research has therefore yet to be done on a better objective function, either in 3D or 2D.

Our method would also greatly benefit from using appearance cues. So far, the decision is based purely on geometrical considerations. Using appearance to improve robustness can be incorporated in our framework as we use an image-based 2D objective function.

CHAPTER

5

CONCLUSION AND PERSPECTIVES.

We have presented a complete pipeline on how to measure behaviours of rodents from videos. From the acquisition of morphologies and motion to tracking, we have broached various fields such as image processing, computer vision and computer graphics. In particular, we have proposed a single animation model of rodents combining both the skeleton and the skin and a tracking algorithm associated to this model.

Our animation model is based on a new structure : the Oriented Quads Rigging (OQR). OQR is a high-level representation of a shape that produces smooth realistic deformations and is more flexible than skeletons as it is not a hierarchical representation with reduced degrees of freedoms. As such, it allows us to model both full extension and full bending of the spine of rodents with very few primitives, unlike skeletons, which would require many joints to produce a smooth bend of the skin.

Thanks to this model, our tracking algorithm is more robust to noise. Robustness is also increased by the use of manifold learning that gives us a low-dimensional space of plausible OQR configurations in which to search for the solution. For humans, where validation datasets are available, this approach has shown to perform well compared to state-of-the-art techniques.

Of course, there is room for improvement and future work. The first thing that one can notice is that the quality of our results is highly dependent on the quality of the input data. Indeed, in a controlled environment such as treadmill locomotion where the cameras can be set up to have the animal take most of the field of view, markerless tracking performs correctly. However, in unconstrained environments such as explo-

ration sequences or parabolic flights, the animal is a small part of the field of view and the precision decreases. As a result, we are unable to properly track the legs but can robustly track the body and the head. Consequently, one can think that a multi-resolution scheme can improve the performance both in terms of results and computation time. A hierarchy of latent spaces as presented in [PR11b] is something worth exploring. One could also create a multi-resolution OQR where, given a resolution, only a subset of oriented quads would be active, whereas the others would be inferred from the active ones.

In my opinion, the most promising path to improve precision would be to add feature tracking to help and improve the results of latent space tracking. Techniques developed for object recognition could be used to track features of interest such as the nose, the ears and the feet. These features could then be used to derive tracking constraints as the centers of colour patches are in [WP09].

One might say that using learning is a limitation of our method. Indeed, in the context of animal studies where databases of movements are not available, manual labeling is required. Because of the tediousness of the process, a minimal subset of examples is often used as learning samples. Unfortunately, the smaller the learning samples, the less all-encompassing the latent space is. To make the labeling process less tedious and more effective, one could imagine that the tracking results and its quality as measured by the objective function could be used to determine which frames are not tracked correctly. The user could then determine if the tracking fails because of the input data (e.g. the head is not reconstructed on the visual hull) or because the posture can not be reached with the current manifold. In the latest case, the user could label this frame and re-try the tracking. However, this points to a drawback of using GP-LVM for manifold learning. When using GP-LVM, adding a learning sample means re-learning the entire latent space from scratch. As it is a computationally expensive process, manually labeling and re-learning could not be done interactively.

Finally, building a set-up with both 8 standard cameras and 2 X-ray devices is necessary to fully measure the accuracy and precision of our method. In such a set-up, we could surgically implant markers on the bones, perform markerless tracking and be able to compare in 3D the tracked positions of the skeletal structures to the real ones.

In retrospect, this work has allowed us to explore if and when existing techniques fail when dealing with the difficulty of real-life inputs, may it be the shape and deformation rodents can assume or the capturing set-ups with few low-resolution grey-level cameras and to propose new solutions to overcome some of these issues as well as leads for a thorough validation procedure. A powerful incentive for this research has been the interdisciplinarity collaboration that has allowed me to participate in projects with fallouts outside of computer science.

BIBLIOGRAPHY

[4dv] *4d view solutions*.

[AWM+11] S. Amiri, D.R. Wilson, B.A. Masri, G. Sharma, and C. Anglin, *A novel multi-planar radiography method for three dimensional pose reconstruction of the patellofemoral and tibiofemoral joints after arthroplasty.*, J Biomech (2011).

[BBDR05] Serge Belongie, Kristin Branson, Piotr Dollár, and Vincent Rabaud, *Monitoring animal behavior in the smart vivarium*, Measuring Behavior (Wageningen, NL), 2005, pp. 70–72.

[BBG+10] E.L. Brainerd, D.B. Baier, S.M. Gatesy, T.L. Hedrick, K.A. Metzger, S.L. Gilbert, and J.J. Crisco, *X-ray reconstruction of moving morphology (xromm): precision, accuracy and applications in comparative biomechanics research.*, J Exp Zool A Ecol Genet Physiol (2010).

[BC10] Jay M. Bauman and Young-Hui Chang, *High-speed x-ray video demonstrates significant skin movement errors with standard optical kinematics during rat locomotion*, Journal of Neuroscience Methods **186** (2010), no. 1, 18 – 24.

[BKZ+07] M.J. Bey, S.K. Kline, R. Zauel, T.R. Lock, and P.A. Kolowich, *Measuring dynamic in-vivo glenohumeral joint kinematics: Technique and preliminary results.*, J Biomech (2007).

[Bra00] G. Bradski, *The OpenCV Library*, Dr. Dobb's Journal of Software Tools (2000).

[Bro71] Duane C. Brown, *Close-range camera calibration*, PHOTOGRAMMETRIC ENGINEERING **37** (1971), no. 8, 855–866.

[BZBT06] M.J. Bey, R. Zauel, S.K. Brock, and S. Tashman, *Validation of a new model-based tracking technique for measuring three-dimensional, in vivo glenohumeral joint kinematics.*, J Biomech Eng **128** (2006), no. 4, 604–9.

[CBI10] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic, *Probabilistic deformable surface tracking from multiple videos*, ECCV, 2010.

[CFM+08] P.A. Couto, V.M. Filipe, L.G. Magalhães, J.E. Pereira, L.M. Costa, P. Melo-Pinto, J. Bulas-Cruz, A.C. Maurício, S. Geuna, and A.S.P. Varejão, *A comparison of two-dimensional and three-dimensional techniques for the determination of hindlimb kinematics during treadmill locomotion in rats following spinal cord injury.*, J Neurosci Methods (2008).

[CGMA10] S. Corazza, E. Gambaretto, L. Mundermann, and T.P. Andriacchi, *Automatic generation of a subject-specific model for accurate markerless motion capture and biomechanical applications*, Biomedical Engineering, IEEE Transactions on **57** (2010), no. 4, 806 –812.

[CKBH00] Kong Man Cheung, Takeo Kanade, J.-Y. Bouguet, and M. Holler, *A real time system for robust 3d voxel reconstruction of human motions*, Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00), vol. 2, June 2000, pp. 714 – 720.

[CMG+10] Stefano Corazza, Lars Mündermann, Emiliano Gambaretto, Giancarlo Ferrigno, and Thomas Andriacchi, *Markerless motion capture through visual hull, articulated icp and subject specific model generation*, International Journal of Computer Vision **87** (2010), 156–169, 10.1007/s11263-009-0284-3.

[DMKH05] Douglas A. Dennis, Mohamed R. Mahfouz, Richard D. Komistek, and William Hoff, *In vivo determination of normal and anterior cruciate ligament-deficient knee kinematics*, Journal of Biomechanics **38** (2005), no. 2, 241 – 253, <ce:title>Knee Mechanics: An Update of Theoretical and Experimental Analyses</ce:title>.

[DSG+04] L.E. DeFrate, H. Sun, T.J. Gill, H.E. Rubash, and G. Li, *In vivo tibiofemoral contact analysis using 3d mri-based knee models.*, J Biomech **37** (2004), no. 10, 1499–504.

[FB03] Jean-Sébastien Franco and Edmond Boyer, *Exact polyhedral visual hulls*, British Machine Vision Conference (BMVC'03) (Norwich, Royaume-Uni), vol. 1, 2003, pp. 329–338 (Anglais).

[FKR05] Michael S. Floater, Géza Kós, and Martin Reimers, *Mean value coordinates in 3D*, Computer Aided Geometric Design **22** (2005), no. 7, 623–631.

[FPC+06] Vitor M. Filipe, José E. Pereira, Luís M. Costa, Ana C. Maurício, Pedro A. Couto, Pedro Melo-Pinto, and Artur S.P. Varejão, *Effect of skin movement on the analysis of hindlimb kinematics during treadmill locomotion in rats*, Journal of Neuroscience Methods **153** (2006), no. 1, 55 – 61.

[GBJD10] S.M. Gatesy, D.B. Baier, F.A. Jenkins, and K.P. Dial, *Scientific rotoscoping: a morphology-based method of 3-d motion analysis and visualization.*, J Exp Zool A Ecol Genet Physiol (2010).

[GMHP04] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović, *Style-based inverse kinematics*, ACM Trans. Graph. **23** (2004), 522–531.

[GRBS10] Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel, *Optimization and filtering for human motion capture*, International Journal of Computer Vision **87** (2010), 75–92, 10.1007/s11263-008-0173-1.

[GRP10] Benjamin Gilles, Lionel Revéret, and Dinesh Pai, *Creating and animating subject-specific anatomical models*, Comput. Graph. Forum **29** (2010), no. 8, 2340–2351.

[GSdA+09] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, *Motion capture using joint skeleton tracking and surface estimation*, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, june 2009, pp. 1746 –1753.

[Hed08] Tyson L Hedrick, *Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems*.

[JBS06] M.W. Jones, J.A. Baerentzen, and M. Sramek, *3d distance fields: a survey of techniques and applications*, Visualization and Computer Graphics, IEEE Transactions on **12** (2006), no. 4, 581 –599.

[JMD+07] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki, *Harmonic coordinates for character articulation*, ACM Transactions on Graphics (TOG). Proccedings of ACM SIGGRAPH **26** (2007), no. 3.

[JSW05] Tao Ju, Scott Schaefer, and Joe Warren, *Mean Value Coordinates for Closed Triangular Meshes*, ACM Transaction on Graphics (TOG), Proceedings of ACM SIGGRAPH **24** (2005), no. 3.

[KBKL09] A. Kolb, E. Barth, R. Koch, and R. Larsen, *Time-of-Flight Sensors in Computer Graphics*, Eurographics 2009 - State of the Art Reports (CH-1288 Aire-la-Ville) (M. Pauly and G. Greiner, eds.), Eurographics Association, Eurographics, March 2009, pp. 119–134.

[KCZO07] Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O'Sullivan, *Skinning with dual quaternions*, Symposium on Interactive 3D Graphics and Games (i3D) (2007).

[Lau94] A. Laurentini, *The visual hull concept for silhouette-based image understanding*, IEEE Trans. Pattern Anal. Mach. Intell. **16** (1994), no. 2, 150–162.

[Law05] N. D. Lawrence, *Probabilistic non-linear principal component analysis with gaussian process latent variable models*, Machine Learning Research **6** (2005), 1783 – 1816.

[LC87] William E. Lorensen and Harvey E. Cline, *Marching cubes: A high resolution 3d surface construction algorithm*, COMPUTER GRAPHICS **21** (1987), no. 4, 163–169.

[LCF00] J. P. Lewis, Matt Cordner, and Nickson Fong, *Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation*, Proceedings of the 27th annual conference on Computer graphics and interactive techniques (New York, NY, USA), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 165–172.

[LLCO08a] Yaron Lipman, David Levin, and Daniel Cohen-Or, *Green coordinates*, ACM Transactions on Graphics (TOG). Proceedings of ACM SIGGRAPH **27** (2008), no. 3.

[LLCO08b] Yaron Lipman, David Levin, and Daniel Cohen-Or, *Green coordinates*, ACM Trans. Graph. **27** (2008), 78:1–78:10.

[LLCo08c] Yaron Lipman, David Levin, and Daniel Cohen-or, *On the derivation of green coordinates*, Tech. report, 2008.

[LQnC06] Neil D. Lawrence and Joaquin Quiñonero Candela, *Local distance preservation in the gp-lvm through back constraints*, Proceedings of the 23rd international conference on Machine learning (New York, NY, USA), ICML '06, ACM, 2006, pp. 513–520.

[may] *Autodesk maya*.

[MG03] Alex Mohr and Michael Gleicher, *Building efficient, accurate character skins from examples*, ACM SIGGRAPH 2003 Papers (New York, NY, USA), SIGGRAPH '03, ACM, 2003, pp. 562–568.

[PKS+09] A.H. Prins, B.L. Kaptein, B.C. Stoel, J.H.C. Reiber, and E.R. Valstar, *Detecting femur-insert collisions to improve precision of fluoroscopic knee arthroplasty analysis.*, J Biomech (2009).

[PR11a] V.A. Prisacariu and I. Reid, *Nonlinear shape manifolds as shape priors in level set segmentation and tracking*, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, june 2011, pp. 2185 –2192.

[PR11b] Victor Adrian Prisacariu and Ian Reid, *Shared shape spaces*, Computer Vision (ICCV), 2011 IEEE International Conference on, nov. 2011, pp. 2587 –2594.

[Sak78] Hiroaki Sakoe, *Dynamic programming algorithm optimization for spoken word recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing **26** (1978), 43–49.

[SBB10] Leonid Sigal, Alexandru O. Balan, and Michael J. Black, *Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion*, Int. J. Comput. Vision **87** (2010), no. 1-2, 4–27.

[SCD+06] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski, *A comparison and evaluation of multi-view stereo reconstruction algorithms*.

[SF10] Yann Savoye and Jean-Sébastien Franco, *Cage-based Tracking for Performance Animation*, ACCV'10 :the Tenth Asian Conference on Computer Vision (Queenstown, New Zealand), 2010.

[SMCS] Greg Slabaugh, Tom Malzbender, Bruce Culbertson, and Ron Schafer, *A survey of methods for volumetric scene reconstruction from photographs*.

[SS02] Daniel Scharstein and Richard Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, Int. J. Comput. Vision **47** (2002), no. 1-3, 7–42.

[TA03] S. Tashman and W. Anderst, *In-vivo measurement of dynamic joint motion using high speed biplane radiography and ct: application to canine acl deficiency.*, J Biomech Eng **125** (2003), no. 2, 238–45.

[TTB12] Jean-Marc Thiery, Julien Tierny, and Tamy Boubekeur, *Cager: Cage-based reverse engineering of animated 3d shapes*, Computer Graphics Forum **31** (2012), no. 8, 2303–2316.

[UFF06] Raquel Urtasun, David J. Fleet, and Pascal Fua, *3d people tracking with gaussian process dynamical models*, Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (Washington, DC, USA), IEEE Computer Society, 2006, pp. 238–245.

[UPBS08] Luis Unzueta, Manuel Peinado, Ronan Boulic, and Ángel Suescun, *Full-body performance animation with sequential inverse kinematics*, Graph. Models **70** (2008), no. 5, 87–104.

[VBMP08] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović, *Articulated mesh animation from multi-view silhouettes*, ACM Transactions on Graphics **27** (2008), no. 3, 97.

[WAK+10] J.T. Weinhandl, B.S.R. Armstrong, T.P. Kusik, R.T. Barrows, and K.M. O'Connor, *Validation of a single camera three-dimensional motion tracking system.*, J Biomech (2010).

[WB00] Jihong Wang and Timothy J. Blackburn, *X-ray image intensifiers for fluoroscopy*.

[WP09] Robert Y. Wang and Jovan Popović, *Real-time hand-tracking with a color glove*, ACM Transactions on Graphics **28** (2009), no. 3.

[YGvGU11] A. Yao, J. Gall, L. van Gool, and R. Urtasun, *Learning probabilistic non-linear latent variable models for tracking complex activities*, Proceedings of Neural Information Processing Systems (NIPS) (Granada, Spain), December 2011.

[ZB94] Jianmin Zhao and Norman I. Badler, *Inverse kinematics positioning using nonlinear programming for highly articulated figures*, ACM Trans. Graph. **13** (1994), 313–336.

[Zha00] Z. Zhang, *A flexible new technique for camera calibration*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **22** (2000), no. 11, 1330 – 1334.

[ZNA03] Y. Zheng, M.S. Nixon, and R. Allen, *Lumbar spine visualisation based on kinematic analysis from videofluoroscopic imaging.*, Med Eng Phys **25** (2003), no. 3, 171–9.

[ZSCS04] Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz, *Spacetime faces: High-resolution capture for modeling and animation*, SIGGRAPH (2004).

[ZVDH03] Victor Brian Zordan and Nicholas C. Van Der Horst, *Mapping optical motion capture data to skeletal motion using a physical model*, Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (Aire-la-Ville, Switzerland, Switzerland), SCA '03, Eurographics Association, 2003, pp. 245–250.

CONTENTS

LIST OF FIGURES

LIST OF TABLES